

Memorizzazione dei dati ed elaborazione delle interrogazioni

Finora abbiamo considerato modelli dei dati ad alto livello, cioè a livello *logico*. Tale livello è quello corretto per gli utenti della base di dati. Tuttavia, un fattore importante nell'accettazione di un DBMS da parte dell'utente è dato dalle sue prestazioni. Le prestazioni del DBMS dipendono dall'efficienza delle strutture dati e dall'efficienza del sistema nell'operare su tali strutture. Nei capitoli precedenti abbiamo introdotto le nozioni di tupla e relazione e i linguaggi per la loro manipolazione. A livello fisico, tali tuple saranno memorizzate in record di file su memoria secondaria ed una manipolazione efficiente verrà garantita dall'uso di opportune tecniche di elaborazione delle interrogazioni. In particolare, per velocizzare la ricerca dei dati vengono in genere utilizzate particolari strutture di accesso, dette *indici*, che consentono di accedere direttamente ai record corrispondenti alle tuple con un certo valore per un attributo, senza scandire l'intero contenuto del file. La scelta delle strutture di memorizzazione e di indicizzazione più efficienti dipende dal tipo di accessi che si eseguono sui dati. Normalmente, ogni DBMS ha le proprie strategie di implementazione di un modello dei dati; tuttavia, l'utente può influenzare le scelte fatte dal sistema. Le scelte dell'utente a questo riguardo costituiscono la *progettazione fisica* della base di dati e si concretizzano mediante l'utilizzo di opportuni comandi forniti dai DBMS. Una volta determinate le strutture di memorizzazione dei dati ed eventuali strutture ausiliarie di accesso, è compito del DBMS determinare, per ogni operazione di manipolazione dei dati, la strategia più efficiente per eseguirla, date le strutture disponibili. Benché tale processo riguardi tutte le operazioni di manipolazione dei dati, gran parte del processo di ottimizzazione è relativo alle operazioni di interrogazione, poiché il costo delle operazioni è dominato dal costo per il ritrovamento delle tuple, che, essendo specificato mediante condizioni dichiarative, offre notevoli margini di ottimizzazione. Per tale motivo ci concentreremo principalmente sull'elaborazione e l'ottimizzazione delle interrogazioni.

In questo capitolo, dopo aver brevemente introdotto l'architettura generale di un sistema di gestione dati, discuteremo le tecniche utilizzate per la memorizzazione fisica e l'indicizzazione dei dati. Illustreremo poi le principali problematiche nell'elaborazione di interrogazioni e descriveremo brevemente l'attività di progettazione fisica.

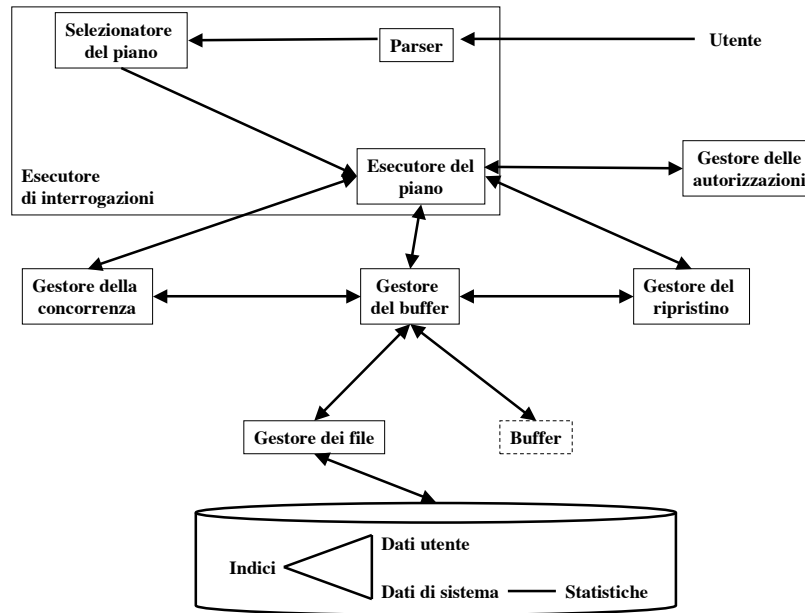


Figura 7.1: Architettura di un DBMS

7.1 Architettura di un DBMS

Le basi di dati memorizzano in modo persistente grosse quantità di dati. La maggior parte delle basi di dati sono memorizzate in maniera permanente su supporti di memorizzazione secondaria (cui ci riferiremo anche come memoria di massa), tipicamente su dischi magnetici. I dati in memoria secondaria non possono essere elaborati direttamente dalla CPU, ma devono essere prima copiati in memoria principale, in un'opportuna area di *buffer*.

Un DBMS deve garantire una gestione efficiente, concorrente, affidabile e sicura dei dati, preservandone inoltre l'integrità. Ciascuno degli aspetti precedenti è supportato nel DBMS da specifiche componenti, o sottosistemi, che complessivamente rappresentano l'architettura del sistema, illustrata graficamente nella Figura 7.1. Nella Figura 7.1, le componenti funzionali sono indicate da rettangoli, mentre il disco (memoria secondaria) è indicato con un cilindro e le aree in memoria principale con un rettangolo tratteggiato. Un DBMS è quindi costituito da diverse componenti funzionali che includono:

- **gestore dei file**, che gestisce l'allocazione dello spazio su disco e le strutture dati usate per rappresentare le informazioni memorizzate su disco;

- **gestore del buffer**, responsabile del trasferimento delle informazioni tra disco e memoria principale;
- **esecutore di interrogazioni**, responsabile dell'esecuzione delle richieste utente e costituito da:
 - **parser**, che traduce i comandi del DDL e del DML in un formato interno (*parse tree*);
 - **selezionatore del piano**, che stabilisce il modo più efficiente di processare una richiesta utente;
 - **esecutore del piano**, che processa le richieste utente in accordo al piano di esecuzione selezionato;
- **gestore delle autorizzazioni**, che controlla che gli utenti abbiano gli opportuni diritti di accesso ai dati;
- **gestore del ripristino**, che assicura che la base di dati rimanga in uno stato consistente a fronte di cadute o malfunzionamenti del sistema;
- **gestore della concorrenza**, che assicura che le esecuzioni concorrenti di processi procedano senza conflitti.

Come evidenziato nella Figura 7.1, il DBMS memorizza, oltre ai dati utente, anche dati di sistema (quali informazioni sullo schema della base di dati e sulle autorizzazioni), strutture ausiliarie di accesso a tali dati (indici) e dati statistici (quali il numero di tuple in una relazione), utilizzati dal selezionatore del piano per determinare la migliore strategia di esecuzione. Tutte le diverse componenti accedono a dati utente e/o di sistema nello svolgimento delle loro funzioni (ad esempio, il selezionatore del piano accede ai dati statistici ed il gestore delle autorizzazioni accede alle autorizzazioni), interagendo quindi con il gestore del buffer, anche se, per semplicità, nella Figura 7.1 alcune frecce sono state omesse.

In questo capitolo, introdurremo innanzitutto brevemente la gestione dei dati di sistema. Ci concentreremo poi sulle problematiche inerenti alla memorizzazione dei dati e l'ottimizzazione di interrogazioni, che concorrono alla realizzazione di una gestione efficiente dei dati, introducendo, quindi, il gestore dei file, il gestore del buffer e l'esecutore di interrogazioni. Il gestore delle autorizzazioni, responsabile dell'integrità e della riservatezza dei dati, ha come componente principale un meccanismo di controllo dell'accesso quali quelli che verranno discussi nel Capitolo 9. I gestori di concorrenza e ripristino, garanti, rispettivamente, dell'accesso concorrente ai dati e della loro affidabilità, saranno discussi nel Capitolo 8.

Cataloghi di sistema. Un DBMS descrive i dati che gestisce, incluse le informazioni sullo schema della base di dati e gli indici, tramite *meta-dati*. Tali meta-dati sono memorizzati in relazioni speciali, dette *cataloghi* di sistema, e sono utilizzati dalle diverse componenti del DBMS. In tali cataloghi vengono innanzitutto memorizzate informazioni di schema. Ad esempio, per ogni relazione vengono

nomeA	nomeR	pos	tipo
nomeA	Attributi	1	VARCHAR(20)
nomeR	Attributi	2	VARCHAR(20)
pos	Attributi	3	INTEGER
tipo	Attributi	4	VARCHAR(10)
titolo	Film	1	VARCHAR(30)
regista	Film	2	VARCHAR(20)
anno	Film	3	DECIMAL(4)
genere	Film	4	CHAR(15)
valutaz	Film	5	NUMERIC(3,2)
codCli	Cliente	1	DECIMAL(4)
...
dataRest	Noleggio	4	DATE

Figura 7.2: Catalogo relativo allo schema della videoteca

mantenuti il nome della relazione, il nome e il tipo di ciascuno degli attributi, il nome di ciascun indice definito sulla relazione, gli eventuali vincoli definiti sulla relazione. Analogamente, per ogni indice vengono memorizzati il nome, il tipo, gli attributi su cui è definito; per ogni vista vengono memorizzati il nome e la definizione. Nei cataloghi vengono inoltre memorizzate statistiche sulle relazioni e sugli indici (vedi Paragrafo 7.4.2), oltre ad informazioni sugli utenti del DBMS e sui loro diritti di accesso (vedi Capitolo 9).

Un aspetto elegante di un DBMS relazionale è che i cataloghi di sistema sono essi stessi relazioni. Ad esempio, le informazioni sulle relazioni e sulle viste saranno contenute in un catalogo avente come attributi: il nome della relazione, il nome dell'utente creatore, il tipo (relazione/vista) ed il numero di attributi. In un altro catalogo verranno ad esempio memorizzate le informazioni sugli attributi delle relazioni e delle viste. Tale relazione conterrà una tupla per ogni attributo di ogni relazione o vista, con attributi: il nome dell'attributo, il nome della relazione o vista a cui l'attributo appartiene, la posizione ordinale dell'attributo tra gli attributi della stessa relazione o vista, il tipo dell'attributo. Un esempio di tale relazione, che supponiamo essere di schema `Attributi(nomeA,nomeR,pos,tipo)`, relativamente alla base di dati della videoteca, è presentato nella Figura 7.2. Notiamo come la relazione contenga anche le informazioni relative agli attributi della relazione stessa.

Il fatto che i cataloghi di sistema siano relazioni permette di interrogarli come tutte le altre relazioni e di applicare ad essi tutte le tecniche per l'implementazione e la gestione di relazioni. I DBMS esistenti utilizzano schemi di catalogo differenti, ma le informazioni contenute sono pressapoco le stesse.

7.2 Memorizzazione dei dati e gestione del buffer

In questo paragrafo introdurremo le nozioni alla base della memorizzazione fisica dei dati su disco e discuteremo due aspetti cruciali nel minimizzare il costo di