



NOME CORSO

Fintech Software Developer

Unità Formativa (UF): Basi di dati SQL

Docente: Durando Giulio

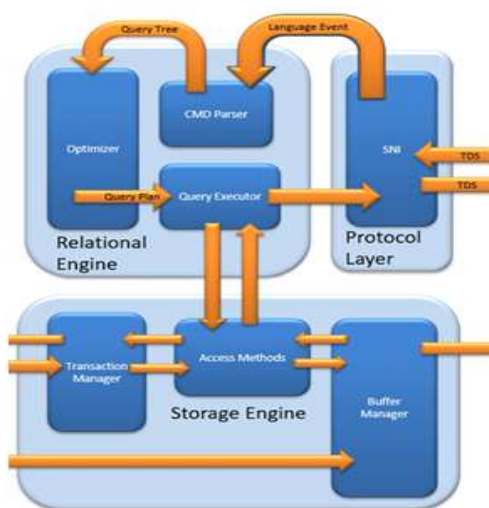
Titolo argomento: Architettura di Microsoft SQL Server



Architettura di SQL Server

SQL Server è un sistema di gestione di database relazionali (RDBMS). È stato sviluppato e commercializzato da Microsoft nel 1989, come SQL Server 1.0, e scritto in C/C++. Inizialmente prima del 2016 era supportato solo in ambiente Windows ma successivamente è stato reso disponibile per Linux, Microsoft Windows Server, Microsoft Windows.

Il server SQL è anche legato a Transact-SQL o T-SQL. Fondamentalmente, T-SQL è un insieme di estensioni di programmazione di Microsoft che aggiunge diverse funzionalità allo Structured Query Language (SQL), incluso il controllo delle transazioni, la gestione delle eccezioni e degli errori, l'elaborazione delle righe e le variabili dichiarate.



Microsoft SQL Server è un'architettura client-server. Il processo di MS SQL Server inizia con l'invio di una richiesta da parte dell'applicazione client. SQL Server accetta processi e risponde alla richiesta con i dati elaborati.

Come illustra il diagramma a lato, ci sono tre componenti principali nell'architettura di SQL Server:

- **Protocol Layer (Livello di protocollo)**
- **Relational Engine (Motore relazionale)**
- **Storage Engine (Motore di archiviazione)**

Protocol Layer - Server Network Interface (SNI)

SNI è il livello di protocollo che fornisce la connessione tra client e server. SQL Server supporta i seguenti 4 protocolli.

- **Shared Memory (Memoria condivisa):** Il protocollo utilizzato quando il client e il server si trovano sulla stessa macchina. Non si può configurare, si può solo disabilitarlo. Se si sta utilizzando il server in cui è installato il server sql come client, per impostazione predefinita il server sql tenta di connettersi all'istanza utilizzando questo protocollo.
- **Named Pipes:** Questo protocollo viene utilizzato quando Client e Server si trovano sulla stessa LAN. Utilizza la porta TCP 445. Può essere utilizzato in ambienti in cui non è presente il protocollo TCP/IP.
- **TCP/IP:** Il protocollo più utilizzato e accettato in tutto il mondo. Quando un client si connette a un server sql, utilizza un IP o un nome server e una porta (ad esempio 10.56.43.23,1433). La porta TCP predefinita di SQL Server è 1433.
- **Virtual Interface Adapter (VIA):** Non è possibile visualizzare questo protocollo in Gestione configurazione SQL Server. Sebbene sia un protocollo ad alte prestazioni, non è preferito perché richiede hardware aggiuntivo nel client e nel server.

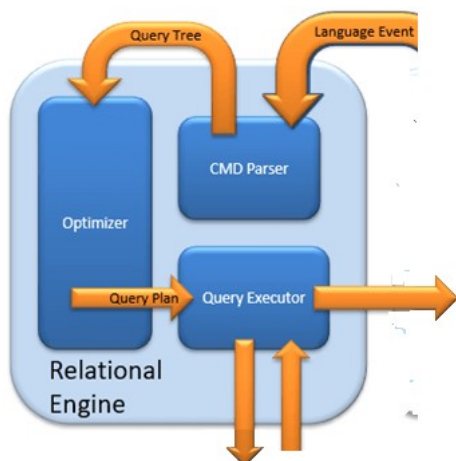


Motore di database:

E' il componente principale dell'architettura di SQL Server che serve per l'archiviazione, l'elaborazione e la protezione dei dati. SQL Server supporta un massimo di 50 istanze di Motore di database su un singolo computer. Fornisce un accesso controllato e un'elaborazione rapida delle transazioni per soddisfare i requisiti delle applicazioni che consumano più dati nelle aziende. Anche gli oggetti di database come stored procedure, viste e trigger vengono creati ed eseguiti tramite Motore di database.

Internamente, è costituito da un motore relazionale per l'elaborazione delle query e da un motore di archiviazione per la gestione di file di database, pagine, indici, ecc.

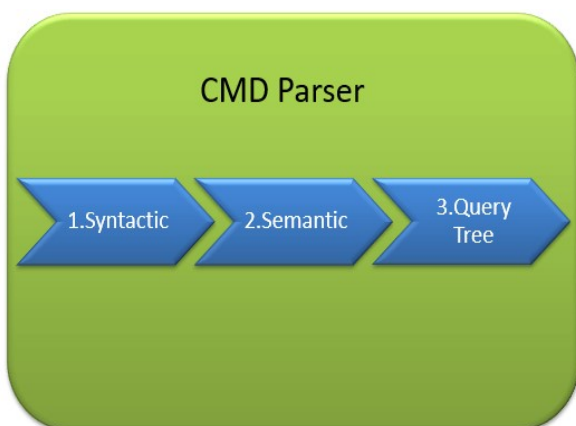
Motore relazionale (Relational Engine):



Il compito del motore relazionale è elaborare le query e quindi è anche chiamato Query Processor. Ma alcune delle attività del motore relazionale possono includere anche l'elaborazione di query, la gestione della memoria, la gestione di thread e attività, la gestione del buffer e l'elaborazione di query distribuite. Di solito richiede i dati dal motore di archiviazione per una determinata query di input ed elabora l'output sulla base di ciò. Ci sono 3 componenti principali del motore relazionale. Questi sono:

- **CMD Parser**
- **Optimizer**
- **Query Executor**

CMD Parser



E' il primo componente del motore relazionale a ricevere i dati della query. Controlla principalmente la query per errori sintattici e semantici. E infine genera un albero di query.



Verifica sintattica:

- Come ogni altro linguaggio di programmazione, anche MS SQL ha un set predefinito di parole chiave. Inoltre, SQL Server dispone di una propria grammatica che SQL Server comprende (SELECT, INSERT, UPDATE e molte altre appartengono agli elenchi di parole chiave predefinite di MS SQL).
- CMD Parser esegue il controllo sintattico. Se l'input degli utenti non segue queste regole di sintassi linguistica o grammaticale, restituisce un errore.

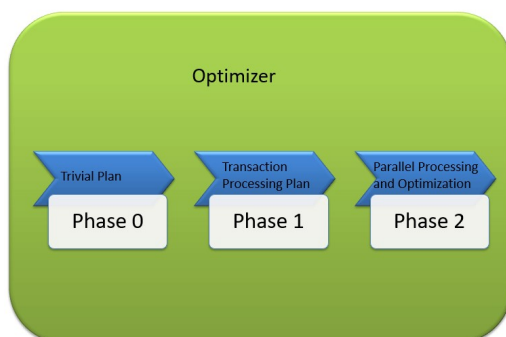
Controllo semantico:

- Questo controllo viene eseguito da Normalizer .
- Nella sua forma più semplice, controlla se il nome della colonna, il nome della tabella oggetto di query esiste in Schema. E se esiste, lo associa alla Query. Questo è anche noto come Binding .
- La complessità aumenta quando le query utente contengono VIEW. Normalizer esegue la sostituzione con la definizione della vista memorizzata internamente e molto altro.

Query tree:

- Questo passaggio genera un albero di esecuzione diverso in cui è possibile eseguire la query.
- Si noti che tutti i diversi alberi hanno lo stesso output desiderato.

Optimizer



Il compito principale dell'ottimizzatore è trovare il piano di esecuzione più economico, non il migliore e conveniente. L'ottimizzazione viene eseguita principalmente per i comandi DML (SELECT, INSERT, UPDATE, DELETE) e non per tutte le query. L'obiettivo finale è ridurre al minimo il tempo di esecuzione della query.

Fase 0: Search for Trivial Plan (ricerca di un piano banale):

- Questa è anche nota come **fase di pre-ottimizzazione** .
- In alcuni casi, potrebbe esserci un solo piano pratico e praticabile, noto come piano banale. Non è necessario creare un piano ottimizzato. Il motivo è che un'ulteriore ricerca porterebbe allo stesso tempo di esecuzione. Anche questo con il costo aggiuntivo della ricerca di un piano ottimizzato che non era affatto richiesto.
- Se non viene trovato alcun piano Trivial, inizia la 1ª Fase.

Fase 1: Search for Transaction processing plans (ricerca dei piani di elaborazione delle transazioni)

- Ciò include la ricerca di un **piano semplice e complesso** .
- Ricerca semplice del piano: i dati passati della colonna e dell'indice coinvolti nella query verranno utilizzati per l'analisi statistica. Questo di solito consiste ma non è limitato a un indice per tabella.
- Tuttavia, se il piano semplice non viene trovato, viene cercato il piano più complesso. Implica l'indice multiplo per tabella.



Fase 2: Parallel Processing and Optimization (elaborazione parallela e ottimizzazione).

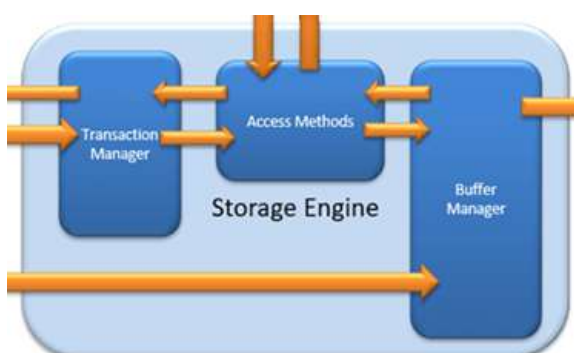
- Se nessuna delle strategie di cui sopra funziona, Optimizer cerca le possibilità di elaborazione parallela. Ciò dipende dalle capacità di elaborazione e dalla configurazione della Macchina.
- Se ciò non è ancora possibile, inizia la fase di ottimizzazione finale. Ora, l'obiettivo finale dell'ottimizzazione è trovare tutte le altre possibili opzioni per eseguire la query nel modo migliore. Gli algoritmi della fase di ottimizzazione finale sono di proprietà di Microsoft.

Esecutore di query (query executor)

Chiama il metodo di accesso. Fornisce un piano di esecuzione per il recupero dei dati della logica richiesta per l'esecuzione. Dopo che i dati sono stati ricevuti da Storage Engine, il risultato passa al livello Protocollo. Infine, i dati vengono inviati all'utente finale.

Motore di archiviazione (Storage Engine)

E' principalmente responsabile dell'archiviazione e del recupero dei dati dai sistemi di archiviazione come dischi e SAN. Ha i seguenti tre componenti principali



- **Access Methods:** determina se la query è un'istruzione Select o Non-Select. E quindi invoca buffer e transfer Manager di conseguenza.
- **Buffer Manager:** gestisce le funzioni principali per Plan Cache, Data Parsing e Dirty Page.
- **Transaction Manager:** gestisce le transazioni non selezionate con l'aiuto di Log e Lock Manager. Inoltre, promuove l'implementazione della registrazione Write Ahead e dei writer Lazy.

Access Method

Funge da interfaccia tra l'esecutore della query e il Gestore buffer/Registri delle transazioni. Il metodo di accesso stesso non esegue alcuna esecuzione.

La prima azione è determinare se la query è:

- *Select Statement (DDL)*
- *Non- Select Statement (DDL & DML)*

A seconda del risultato, il metodo di accesso effettua le seguenti operazioni:

- Se la query è **DDL**, (istruzione SELECT), la query viene passata al Buffer Manager per un'ulteriore elaborazione.
- Se la query è **DDL**, (NON-SELECT statement), la query viene passata a Transaction Manager. Ciò include principalmente l'istruzione UPDATE.



Buffer Manager

Buffer manager gestisce le funzioni principali per i moduli seguenti:

- Plan Cache (pianifica la cache)
- Data Parsing: Buffer cache & Data storage (cache del buffer e archiviazione dei dati)
- Dirty Page

Plan Cache

- **Existing Query plan:** il gestore del buffer verifica se il piano di esecuzione è presente nella cache del Plan Cache. Se sì, vengono utilizzate la cache del piano di query e la cache dei dati associata.
- **First time Cache plan:** da dove viene la cache del piano esistente?
Se il piano di esecuzione della query per la prima volta è in esecuzione ed è complesso, è opportuno archiviarlo nel Plan cache. Ciò garantirà una disponibilità più rapida la prossima volta che SQL Server riceverà la stessa query. Quindi il piano di esecuzione della query viene archiviato se la query viene eseguita per la prima volta.

Data Parsing: Buffer cache & Data Storage

Buffer manager fornisce l'accesso ai dati richiesti. Di seguito sono possibili due approcci a seconda che i dati siano presenti o meno nella cache dei dati:

- **Buffer Cache – Soft Parsing:** Buffer Manager cerca i dati nel buffer nella cache dei dati. Se presenti, questi dati vengono utilizzati da Query Executor. Ciò migliora le prestazioni poiché il numero di operazioni di I/O viene ridotto durante il recupero dei dati dalla cache rispetto al recupero dei dati dall'archivio dati.
- **Data Storage – Hard Parsing:** Se i dati non sono presenti in Buffer Manager, i dati richiesti vengono cercati in Data Storage. Se memorizza anche i dati nella cache dei dati per un uso futuro.
- **Dirty Page:** sono le pagine nel buffer di memoria che hanno modificato i dati, ma i dati non vengono spostati dalla memoria al disco.

Transaction Manager

Gestisce le transazioni non selezionate con l'aiuto di Log manager e Lock Manager. Inoltre, promuove l'implementazione della registrazione Write Ahead e dei writer Lazy. Transaction Manager viene richiamato quando il metodo di accesso determina che la Query è un'istruzione Non-Select.

Log Manager

- Log Manager tiene traccia di tutti gli aggiornamenti effettuati nel sistema tramite i registri nei registri delle transazioni.
- I registri hanno il numero di sequenza dei registri con l'ID transazione e il record di modifica dei dati.

Lock Manager

- Durante la transazione, i dati associati in Data Storage sono nello stato di blocco. Questo processo è gestito da Lock Manager.
- Questo processo garantisce la **coerenza e l'isolamento dei dati**. Conosciuto anche come proprietà **ACID**.



Execution Process

- Log Manager avvia la registrazione e Lock Manager blocca i dati associati.
- La copia dei dati viene mantenuta nella cache del buffer.
- La copia dei dati che dovrebbero essere aggiornati viene mantenuta nel buffer di registro e tutti gli eventi aggiornano i dati nel buffer di dati.
- Le pagine che memorizzano i dati sono anche conosciute come **dirty page**.
- **Checkpoint e registrazione write-ahead:** questo processo esegue e contrassegna tutta la pagina da Dirty page su disco, ma la pagina rimane nella cache. La frequenza è di circa 1 esecuzione al minuto. Ma la pagina viene prima inviata alla pagina Dati del file di registro dal registro del buffer. Questo è noto come **Write Ahead Logging**.
- **Lazy Writer:** la pagina sporca può rimanere in memoria. Quando il server SQL osserva un carico enorme ed è necessaria memoria buffer per una nuova transazione, libera pagine sporche dalla cache. Funziona su **LRU** - Algoritmo utilizzato meno di recente per la pulizia della pagina dal pool di buffer al disco.