

MPS Sqlserver

Architettura di SQL Server

L'architettura di SQL Server si basa su due motori principali che lavorano in modo sinergico:

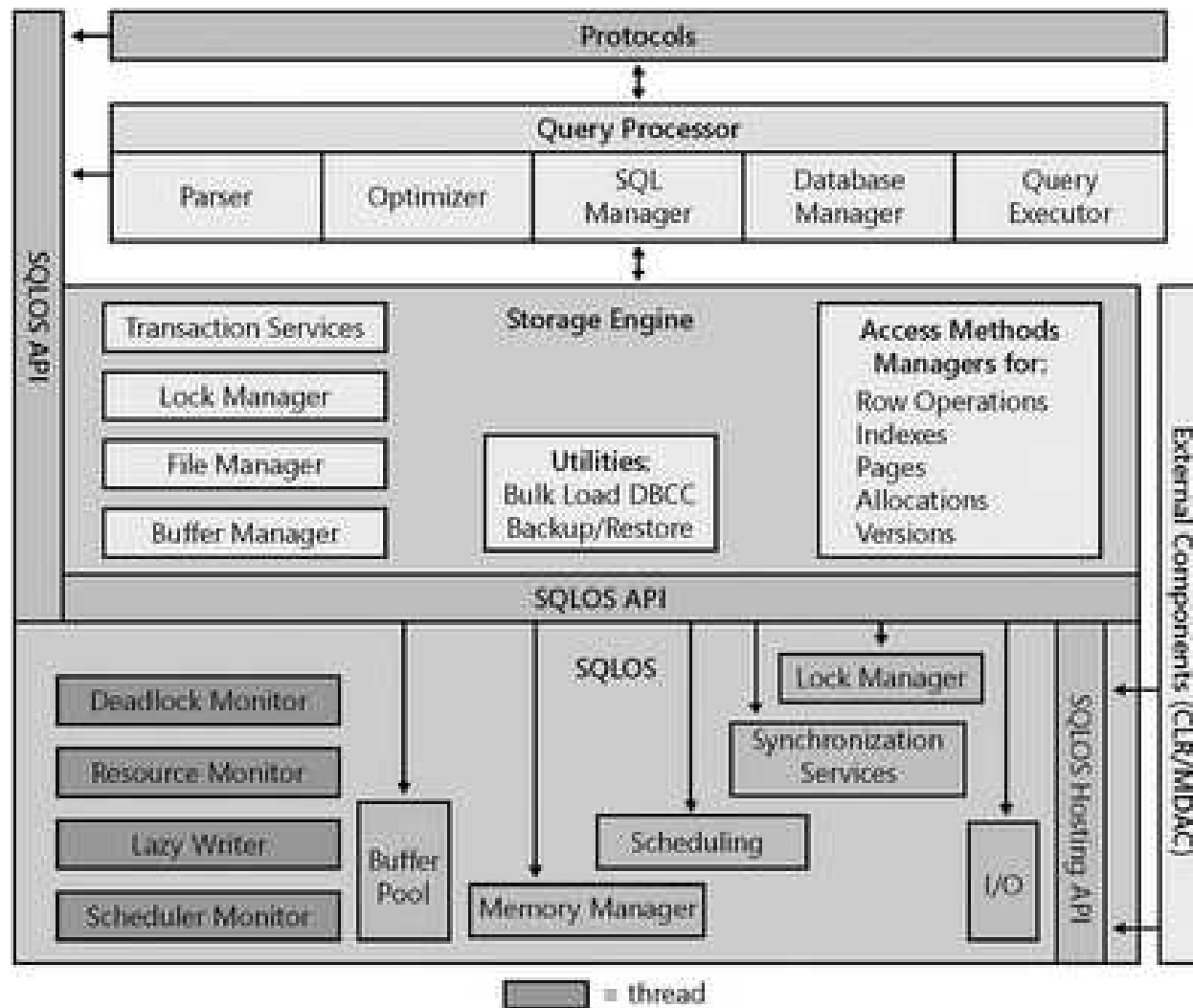
Database Engine (Motore del database): Questo è il nucleo di SQL Server. Gestisce la creazione, la manipolazione e il recupero dei dati. Si compone di:

Relational Engine (Motore relazionale): Accetta le query T-SQL (Transact-SQL) e determina il piano di esecuzione più efficiente. Si occupa dell'ottimizzatore di query, del gestore delle transazioni e della sicurezza.

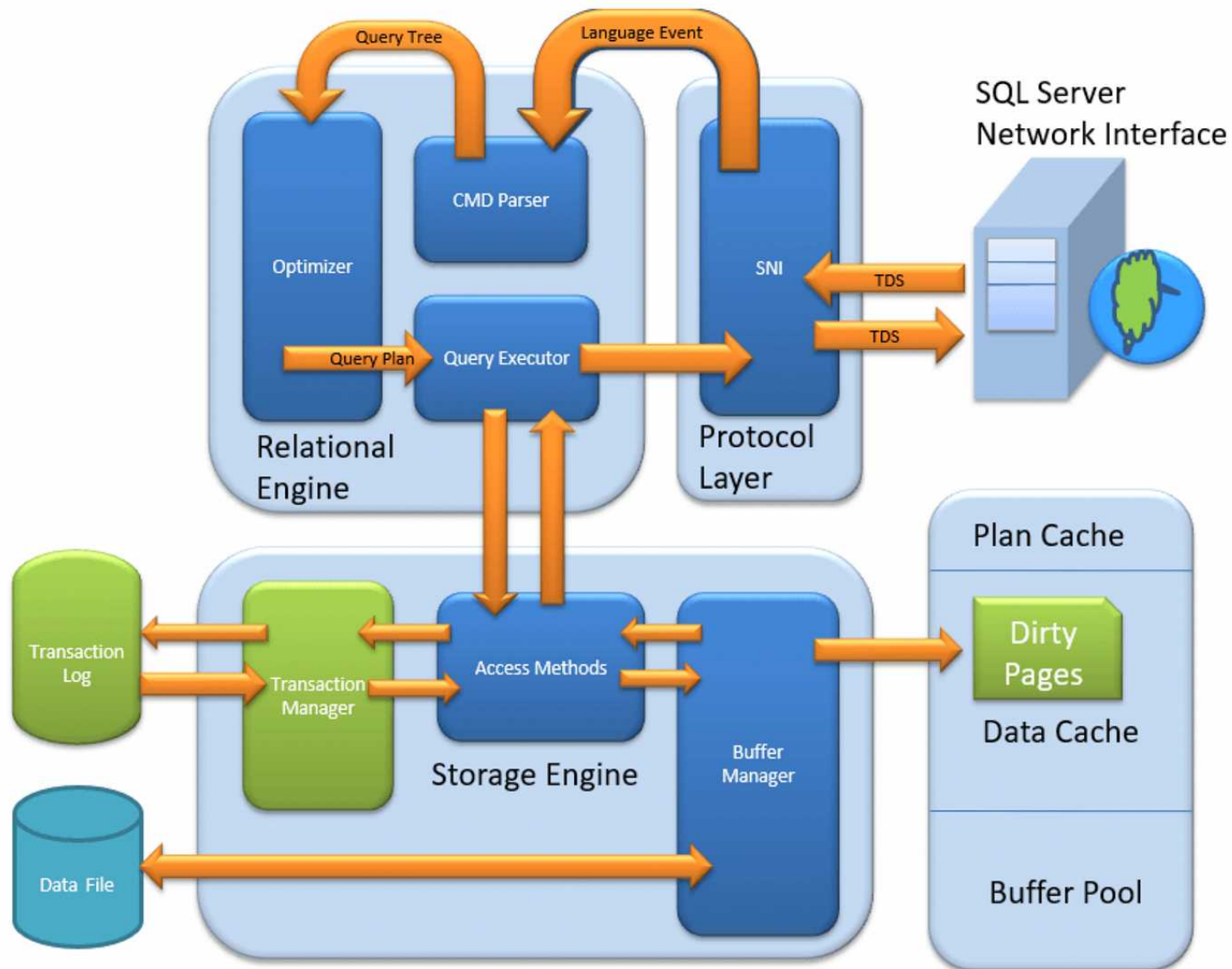
Storage Engine (Motore di archiviazione): Si occupa della gestione fisica dei dati, interagendo direttamente con i dischi per leggere e scrivere le informazioni. Include il gestore dei buffer, il gestore dei file e il gestore dei blocchi.

SQLOS (SQL Server Operating System): Un livello intermedio tra il motore di database e il sistema operativo sottostante (Windows o Linux). Gestisce le risorse di basso livello come la schedulazione dei thread, la gestione della memoria, l'I/O dei file e la sincronizzazione. Funziona come un sistema operativo virtuale per SQL Server, permettendo al Database Engine di funzionare in modo efficiente indipendentemente dal SO.

Architettura



Architettura





Aree di Memoria di SQL Server

SQL Server utilizza la memoria di sistema per ottimizzare le operazioni, riducendo la necessità di leggere e scrivere continuamente su disco, che è un'operazione lenta. Le aree di memoria principali sono:

Buffer Pool (Pool dei buffer): Questa è l'area di memoria più importante e più grande. Contiene le pagine di dati e le pagine di indice lette dai file di database (mdf/ndf). L'obiettivo è mantenere in memoria le pagine più utilizzate per un accesso rapido, minimizzando l'I/O su disco.

Quando una pagina viene richiesta, SQL Server la cerca prima nel Buffer Pool; se non la trova, la carica dal disco.



Aree di Memoria di SQL Server

Plan Cache (Cache dei piani): Qui vengono memorizzati i piani di esecuzione delle query. Quando una query viene eseguita per la prima volta, l'ottimizzatore crea un piano di esecuzione, che viene poi salvato nella Plan Cache.

Se la medesima query viene eseguita di nuovo, SQL Server può riutilizzare il piano esistente, evitando il costoso processo di ricompilazione migliorando così le prestazioni.

Procedure Cache (Cache delle procedure): Questa è un'area specifica del Plan Cache. Memorizza i piani di esecuzione per le stored procedure, i trigger e le funzioni.



Aree di Memoria di SQL Server

TempDB: Sebbene sia un database fisico su disco, una parte significativa delle sue operazioni avviene in memoria. TempDB viene utilizzato per:

- Oggetti temporanei creati dagli utenti (tabelle temporanee, variabili di tabella).

- Operazioni interne al motore di query (ad esempio, ordinamenti complessi o l'uso di cursori).

Workspace Memory (Memoria di lavoro): che viene utilizzata per eseguire operazioni specifiche delle query, come l'ordinamento (SORT), gli hash join e la costruzione di indici. La quantità di memoria allocata a queste operazioni è limitata e gestita separatamente per evitare di monopolizzare tutte le risorse del sistema.

Schema

In SQL Server, uno schema è un container logico che raggruppa oggetti di database, come:

- table,
- view,
- stored procedure,
- index,
- function...

È un livello di astrazione che separa la proprietà e la gestione degli oggetti all'interno di un database, fornendo una struttura organizzativa e di sicurezza.

Schema

Organizzazione: gli schema permettono di raggruppare logicamente oggetti correlati. Ad esempio, in un database per un'azienda, si possono creare schemi separati per diverse aree di business come Vendite, RisorseUmane e Contabilità. Questo evita conflitti di denominazione (es. Vendite.Clienti e RisorseUmane.Clienti possono coesistere).

Sicurezza: Gli schema sono la base per implementare un modello di sicurezza granulare. È possibile assegnare a utenti o ruoli specifici i permessi di accesso a un intero schema, piuttosto che dover gestire i permessi per ogni singolo oggetto.

Schema e Utenti

Proprietario dello Schema: Uno schema è di proprietà di un utente o di un ruolo del database. Il proprietario ha il pieno controllo sugli oggetti all'interno di quello schema.

Schema Predefinito (Default Schema): Ogni utente del database ha uno schema predefinito (default). Quando un utente crea un oggetto (es. una tabella) senza specificare uno schema, l'oggetto viene creato nel suo schema predefinito.

Lo schema predefinito per un nuovo database è **dbo** (database owner).

Questo sistema permette la separazione tra la proprietà degli oggetti (chi li ha creati/gestisce) e i permessi su di essi (chi può leggerli, modificarli, ecc.).

Schema

Creare un nuovo schema:

```
CREATE SCHEMA abc AUTHORIZATION MarioRossi;
```

Creare una tabella all'interno di uno schema:

```
CREATE TABLE abc.Dipendenti (  
    ID INT PRIMARY KEY,  
    Nome VARCHAR(100)  
);
```

Assegnare un permesso a un utente su uno schema:

```
GRANT SELECT ON SCHEMA::Vendite TO GiovanniBianchi;
```

Utilizzatori

Per creare un utente in SQL Server, occorrono le autorizzazioni a due livelli distinti:

Login: È un'entità di sicurezza a **livello di istanza del server**. Permette a una persona o un'applicazione di connettersi a un'istanza di SQL Server. I permessi assegnati a un login si applicano a livello di server (es. creare un database).

Utente: È un'entità di sicurezza a **livello di database**. Un utente è sempre associato a un login. Permette a un login di accedere e interagire con gli oggetti (tabelle, viste, ecc.) all'interno di un database specifico. I permessi assegnati a un utente si applicano solo a quel database.

Login

Ruoli a livello di istanza del server (per il Login):

sysadmin: Questo ruolo è il più potente e ha il controllo totale sull'intera istanza di SQL Server. I membri di questo ruolo possono creare, modificare ed eliminare qualsiasi login e utente.

securityadmin: Questo ruolo è specificamente designato per la gestione della sicurezza, inclusa la creazione e la gestione dei login e dei loro permessi a livello di server.

Utente

Ruoli a livello di database (per l'Utente):

db_owner: Questo ruolo è il più potente a livello di database. I suoi membri hanno il pieno controllo sul database e possono creare, modificare ed eliminare utenti, oltre che gestire i permessi su tutti gli oggetti al suo interno.

db_securityadmin: Questo ruolo è specifico per la gestione della sicurezza all'interno del database e i suoi membri possono gestire permessi, ruoli e utenti.

dbo (Database Owner): È un utente speciale che ha pieni permessi sul database.

In sintesi, per creare un nuovo utente in un database, un utente (o il login a cui è associato) deve avere il permesso CREATE USER su quel database.

Creazione utente

- Un amministratore del server (membro di sysadmin o securityadmin), in genere sa, crea un nuovo login a livello di istanza.
- Un amministratore del database (membro di db_owner o db_securityadmin) crea un nuovo utente nel proprio database, associandolo al login appena creato.

Questo processo in due fasi garantisce la gestione della sicurezza, separando le responsabilità tra l'amministrazione del server e quella dei singoli database.

Creazione utente

Supponendo di essere al primo avvio di sql server e potendo utilizzare *sa*, come creo un nuovo utente ed un nuovo database associato a quell'utente?

vedere procedura **001-CreateUser.sql**

