Risposte Teoria

---1---

Il pool di connessioni consente a un’applicazione di usare una connessione da un pool di connessioni che non devono essere ristabilite per ogni utilizzo. L’utente chiama OPEN() per stabalire una connessione, il pooler cerca una connessione disponibile nel pool. Se disponibile, la restituisce al chiamante anziche aprirne una nuova. Quando l’applicazione chiama CLOSE() sulla connessione, il pooler la restituisce al set di connessioni attive in pool anzichè chiuderlo. Una volta restituita al pool, la connessione è pronta per essere riutilizzata alla successiva chiamata OPEN().

---2---

La connection string è una stringa che fornisce i parametri di inizializzazione per creare una connessione.

È composta da valori in formato Chiave-Valore.

---3---

L’accesso ai dati in ADO.NET è consentita in due modalità:

* In modalità Connessa (Connected Mode)
* In modalità Disconnessa (Disconnected Mode)

La modalità connessa mantiene sempre attiva la connessione al database, i dati sono sempre aggiornati ed è utile per applicazioni con le seguenti caratteristiche: transizioni brevi, pochi accessi paralleli, necessità di dati sempre aggiornati. Le principali classi utilizzate in modalità connessa sono : connection (SqlConnection), Command (SqlCommand), DataReader (SqlDatareader), Parameter (SqlParameter).

La modalità disconnessa non utilizza una connessione permanente al DB ed i dati sono caricati in una memoria centrale. Le modifiche apportate ai dati in memoria centrale non sono immediatamente memorizzate nel DB. Questo tipo di modalità è utile per applicazioni che richiedono numerosi accessi contemporanei al database e che eseguono transizioni di lunga durata (es. Applicazioni Web), indipendentemente dall’ambiente di origine data. Le pricipali classi utilizzate in modalità disconnessa: DataSet, DataTable, DataColumn, DataRow, Constraint, DataRelation.

---4---

Rappresenta un set di comandi SQL e una connessione al database utilizzati per riempire il DataSet e aggiornare l’origine dati.

Il DataAdapter fornisce il ponte tra l'oggetto DataSet e l'origine dati. DataAdapter utilizza oggetti Command per eseguire comandi SQL sull'origine dati per caricare sia DataSet con dati sia per riconciliare le modifiche apportate ai dati nel DataSet con l'origine dati La classe DataAdapter permette di collegare l'origine dati e le classi disconnesse tramite le classi connesse.

---6---

La concorrenza ottimistica viene solitamente usata in ambienti in cui il conflitto per l’uso dei dati non è elevato. Con questo tipo di concorrenza è possibile ottenere un miglioramento delle prestazioni, pochè non è richiesto alcun blocco dei record, operazione che richiede risorse dei server aggiuntive. L’oggetto DataSet è favorito nell’uso della concorrenza ottimistica per attività di lunga durata, quali gestione da remoto dei dati e l’interazione con i dati. La concorrenza pessimistica implica il blocco di righe nell'origine dati per impedire agli altri utenti di apportare modifiche ai dati che possano influire sugli utenti correnti. In un modello pessimistico, quando un utente esegue un'azione che provoca l'applicazione di un blocco, agli altri utenti non sarà consentito eseguire azioni che possano entrare in conflitto con tale blocco, fino a quando tale blocco non verrà rilasciato dal relativo proprietario. Questo tipo di modello viene usato principalmente in ambienti in cui il conflitto per l'uso di dati è elevato e il dispendio di memoria dovuto alla protezione di dati tramite blocchi è inferiore al dispendio relativo all'annullamento di transazioni in caso di conflitti di concorrenza.