

seeMore

Un potente strumento per l'integrazione di dati provenienti da fonti diverse. Fra le possibili applicazioni: la migrazione da un database ad un altro di dati e strutture, la gestione di database non relazionali come se fossero relazionali, l'estensione completa dei livelli di sicurezza nell'accesso ai dati.

LO SCENARIO

Nelle grandi organizzazioni dove sono presenti diverse tecnologie di database (ad es. Oracle, Sybase, DB2, Ingres, file COBOL, ecc.) il problema dell'accesso ai dati in maniera indipendente da dove essi risiedono, per poter essere utilizzati da utenti o da altre applicazioni in maniera integrata, viene risolto attraverso lo sviluppo di software ad hoc che deve essere mantenuto ed aggiornato nel tempo.

Il *Virtual Database Server* di seeMore risolve questo problema agendo da unico punto di accesso e di controllo per tutte le fonti di dati, attraverso il quale i Client interagiscono con qualsiasi database. In effetti, il *Virtual Database* non è un vero e proprio server, ma è piuttosto una combinazione multiplexor e selector di database, con dei connettori ODBC, JDBC e OLEDB sia in input (dove si connettono i database), sia in output (dove si connettono gli utenti / client).

Inoltre, sebbene esso sia un sistema pass-through molto semplice può anche far apparire database non relazionali come fossero relazionali. Ad esempio, è possibile connettere il *Virtual Database Server*, abilitando un client ODBC (o JDBC o OLEDB), ad accedere un database COBOL come fosse un set di database relazionali. La stessa cosa può essere fatta per i database CTtree, C-ISAM, XML e perfino flat file. Il processo è semplicissimo: una volta che i dati sono resi disponibili attraverso i connettori, ogni applicazione, o ambiente di sviluppo o tool di terze parti può accedervi.

Il *Virtual Database Server* accetta immediatamente i più diffusi database: ORACLE, DB2, Microsoft SQL Server, MySQL, SYBASE, INGRES e molti altri: qualsiasi database per cui esistano driver ODBC o JDBC può essere importato in seeMore.

DISEGNARE IL DATABASE

I processi del *Virtual Database Server* sono controllati attraverso il *Virtual Database Designer*, una console grafica che l'amministratore utilizza per importare i dati dalle varie fonti, per definire le liste degli utenti ed i relativi diritti di accesso, costruendo in tal modo la struttura dati che il *Virtual Database Server* presenta ai client. Diversi wizard guidano l'utente e semplificano il lavoro. Fra questi l' SQLJ Wizard consente di scrivere SQLJ Stored Procedure anche senza alcuna esperienza di Java.

I POSSIBILI UTILIZZI

I possibili utilizzi di questa tecnologia vanno oltre la capacità di far apparire database non relazionali come fossero tali:

- ♦ la possibilità di stabilire liste degli utenti abilitati ad accedere al database, in maniera da poter applicare criteri di sicurezza anche in quei casi dove non sarebbe altrimenti possibile. E' possibile stabilire permessi di accesso per i data file COBOL, text file, file C_ISAM o qualsiasi altro database importato nel *Virtual Database Server*;
- ♦ La capacità di costruire nuovi database da singoli elementi di diversi database esistenti;
- ♦ La possibilità di utilizzare dei sinonimi per accedere al database importato nel *Virtual Database Server*, mentre le applicazioni esistenti continuano ad utilizzare i nomi originari.

CARATTERISTICHE PRINCIPALI

- ♦ Numero di cataloghi illimitato
- ♦ Numero illimitato di fonti dati (soggetto a licenza)
- ♦ Wizard per l'importazione rapida delle defizioni delle fonti dati
- ♦ Definizioni delle fonti dati supportate: tabelle (inclusi gli indici) e relazioni, Viste, Stored Procedures.
- ♦ Possibilità di specificare se la fonte dei dati è Read only o Read/Write
- ♦ Aggiunge sicurezza dei dati e degli utenti tramite liste d'accesso (a livello di colonna)
- ♦ Supporto degli schemi
- ♦ Capacità di aggiungere sinonimi alle tabelle, colonne e viste
- ♦ Capacità di aggiungere descrizioni alle tabelle, colonne e viste
- ♦ Capacità di creare tabelle e viste per una fonte dati specifica
- ♦ Supporto di tabelle complesse come quelle del COBOL
- ♦ Supporto per tabelle partitioned attraverso fonti dati multiple
- ♦ Supporto per pseudo colonne consentendo il consolidamento dei dati
- ♦ Supporto per le tabelle mapped
- ♦ Capacità di creare stored Procedures utilizzando SQLJ

SUPPORTO SQL

- ♦ Supporto per tabelle Global e Local temporary
- ♦ Supporto per statement CREATE TABLE AS e LIKE
- ♦ Supporto per CREATE VIEW
- ♦ Supporto per SQL compliant DATE, TIME e TIMESTAMP
- ♦ Supporto per operazioni UNION, INTERSECT, EXCEPT e DISTINCT
- ♦ Supporto per joins INNER, NATURAL, LEFT e OUTER
- ♦ Supporto per ROW negli statement update
- ♦ Supporto per Arrays

ADERENZA AGLI STANDARD

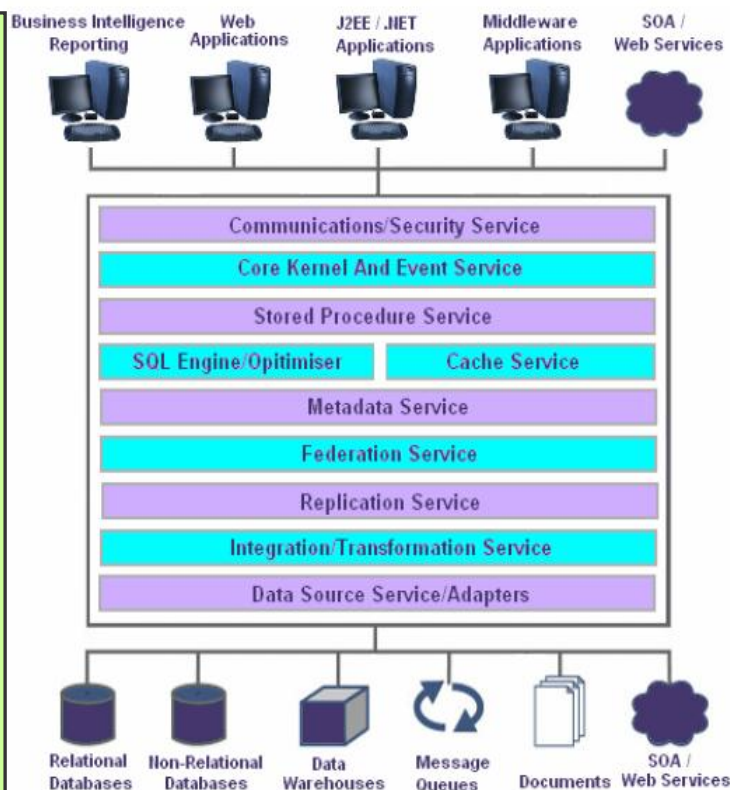
- ♦ Supporto di ANSI SQL-99
- ♦ Supporto delle stored procedures ANSI SQLJ
- ♦ Supporto delle interfaccia ODBC e OLE DB Microsoft
- ♦ Supporto dell'interfaccia JDBC di Sun

SETUP E DEPLOY

- ♦ Looks and feels come un database reale
- ♦ Database Designer per la manutenzione dei Meta Data
- ♦ Sviluppato in Java per la portabilità cross-platform

LE CAPACITA' DI REPLICA

- ♦ Schedulazione della replica di tabelle per partizione
- ♦ Schedulazione della replica di tabelle per vista
- ♦ Capacità di replica supportate: Snapshot, Incremental, Synchronized



LE FONTI SUPPORTATE

- ♦ Oracle Database
- ♦ Oracle (Sleepycat) Berkeley
- ♦ IBM DB2 Database
- ♦ IBM Informix Database
- ♦ Ingres Database
- ♦ Microsoft SQL Server Database
- ♦ Sybase ASE Database
- ♦ MySQL Database
- ♦ Enterprise DB (Postgres)
- ♦ ANTS Data Server
- ♦ Fujitsu NetCobol
- ♦ Micro Focus COBOL
- ♦ CTREE, CISAM and DISAM
- ♦ Flat Files (CSV and Text)
- ♦ XML (using ODBC)
- ♦ Generic (using ODBC)

CONNESSIONE AI DATABASE SERVER

- ♦ Internet sockets communications over TCP/IP
- ♦ MQ Series message system

LE PIATTAFORME SUPPORTATE

- ♦ Windows Server 2003
- ♦ Linux (Redhat, Suse)