

Facoltà di Ingegneria – Sede di Modena
Corso di Laurea in Ingegneria Informatica

Confronto di sistemi per allineamento dati server-PDA

Relatore:

Chiar. mo Prof. Sonia Bergamaschi

Candidato:

S.Ten. Herion Limaj

Correlatore:

Ing. Laura Po

Indice

Capitolo I – Introduzione	5
1.1 Obiettivo della tesi.....	6
1.2 I database per dispositivi mobili.....	6
1.3 Schema di una rete di sincronizzazione server-PDA.....	7
Capitolo II – SQL Server Mobile Edition	9
2.1 Sincronizzazione.....	9
2.1.1 Supporto di più sottoscrizioni.....	9
2.1.2 Supporto multiutente e sincronizzazione.....	10
2.1.3 Stato di avanzamento della sincronizzazione.....	14
2.1.4 Rilevamento a livello di colonna	16
2.2 Motore di archiviazione.....	16
2.2.1 Motore di archiviazione modificato.....	16
2.2.2 Supporto multiutente.....	16
2.2.3 Riutilizzo automatico di pagine vuote.....	18
2.3 Processore delle query.....	18
2.3.1 Ottimizzazione basata sui costi.....	18
2.3.2 Suggerimenti per i piani di esecuzione e per le query.....	19
2.3.3 Nuovo oggetto SqlCeResultSet.....	19
2.4 Integrazione con altri prodotti Microsoft.....	19
2.4.1 Integrazione con SQL Server 2005.....	19
2.4.2 Integrazione con Microsoft Visual Studio 2005.....	22
Capitolo III – SQL Anywhere 11	24
3.1 Tecnologie per la gestione dei dati.....	24
3.1.1 SQL Anywhere Server	24

3.1.2 UltraLite.....	24
3.2 Tecnologie per lo scambio dei dati.....	25
3.2.1 MobiLink-Sincronizzazione.....	25
3.2.2 QAnywhere-scambio messaggi tra gli utenti mobili dentro l'impresa e servizi web mobile.....	28
3.2.3 SQL Remote-replicazione.....	28
3.3 Strumenti di progettazione e di gestione.....	29
Capitolo IV DB2 EVERYPLACE	31
4.1 Query-By-Example.....	32
4.2 DB2 Everyplace Sync Server.....	33
4.3 Come avviene la sincronizzazione.....	36
4.4 Personal Application Builder.....	39
4.5 Conclusioni.....	40
Capitolo V – Oracle Database Lite.....	42
5.1 Mobile Client.....	43
5.2 L'architettura del Oracle Database Lite.....	46
5.3 Mobile Server	46
5.4 Il Mobile Development Kit.....	49
5.5 Configurazione della inizializzazione della applicazione	51
Capitolo VI-Confronto tra i DBMS presi in considerazione.....	53
Capitolo VII-Conclusioni.....	56
Bibliografia.....	57

Elenco delle figure

<i>Figura 1.1:</i> Schema di una rete di sincronizzazione server-PDA	8
<i>Figura 2.1:</i> Query Analyzer 3.0	14
<i>Figura 2.2:</i> Apertura di un database su un PC fisico	20
<i>Figura 2.3:</i> Apertura di un database su un PDA	20
<i>Figura 3.1:</i> Sincronizzazione tramite MobiLink	25
<i>Figura 3.2:</i> Sincronizzazione tramite gli script di sincronizzazione	26
<i>Figura 3.3:</i> Sybase Central	29
<i>Figura 4.1:</i> QBE visualizzazione della tabella.	32
<i>Figura 4.2:</i> QBE command-line processor	33
<i>Figura 4.3:</i> L'architettura della sincronizzazione	34
<i>Figura 4.4:</i> Mobile Devices Administration Center (MDAC)	35
<i>Figura 4.5:</i> Sincronizzazione client – server	36
<i>Figura 4.6:</i> Sincronizzazione Server - client	37
<i>Figura 4.7:</i> Le opzioni di connessione DB2 Everyplace	39
<i>Figura 4.8:</i> DB2 Everyplace Personal Application Builder	40
<i>Figura 5.1:</i> Architettura per il Deployment di Oracle Database Lite usando gli Mobile Servers	42
<i>Figura 5.2:</i> Sincronizzazione dati tra dispositivi mobili e server.	45
<i>Figura 5.3:</i> L'architettura del Oracle Database Lite	46
<i>Figura 5.4:</i> Modello di sviluppo Oracle Lite	50
<i>Figura 5.5:</i> Sincronizzazione con il Mobile Server	52
<i>Figura 5.6:</i> Configurazione Branch del Office Oracle Lite Database	52

Capitolo I

Introduzione

L'uso dei laptop, calcolatori mobili e PDA (personal digital assistant) sta crescendo ed è predisposto a crescere nel futuro vista anche la nascita di un numero cospicuo di applicazioni create apposta per essere installate su dispositivi mobili. Gli analisti non possono dirci con esattezza quale applicazione sarà la più popolare, però è chiaro che una grande percentuale di applicazioni richiederà l'uso del database. Molte applicazioni ideate per PDA richiederanno software e procedure per scaricare informazione da un deposito di informazioni (server) e dovranno prevedere la possibilità di operare su questi dati anche quando si è fuori dalla copertura di rete o disconnessi.

Il crescente successo dei dispositivi mobili permette alle aziende di fornire dati a clienti e dipendenti ovunque essi si trovino. Il potenziale è enorme: un venditore equipaggiato con un PDA dotato del software opportuno può collegarsi al database aziendale per verificare all'istante la disponibilità di un prodotto, lo stato di un ordine o gli ordini effettuati in passato da un cliente. I corrieri possono usare computer palmari per tenere traccia delle consegne e segnalare le modifiche agli ordini in modo da massimizzare l'efficienza della catena distributiva.

L'allineamento dati server-PDA consiste nella sincronizzazione dati tra un database consolidato residente sul server e un database residente sul PDA. I database vengono gestiti da un RDBMS (Relational Database Management System) che è un sistema relazionale per la gestione di basi di dati ed indica un DBMS basato sul modello relazionale introdotto da Edgar F.Codd.

La struttura della tesi è la seguente:

- **Capitolo 1** → Questo capitolo ha lo scopo di introdurre le nozioni di base sui database per dispositivi mobili.
- **Capitolo 2** → In questo capitolo viene presentato SQL Server Mobile il DBMS per palmari fornito dalla Microsoft
- **Capitolo 3** → Questo capitolo descrive SQL Anywhere Studio.
- **Capitolo 4** → Questo capitolo descrive DB2 Everyplace il DBMS per palmari fornito dal IBM
- **Capitolo 5** → In questo capitolo viene presentato Oracle Database Lite
- **Capitolo 6** → In questo capitolo vengono messi in confronto i DBMS presi in considerazione.
- **Capitolo 7** → In questo capitolo vengono presentati, discussi e analizzati i risultati ottenuti nella tesi

1.1 OBIETTIVO DELLA TESI

L'obiettivo di questa tesi è l'analisi e il confronto dei diversi DBMS per dispositivi mobili. L'analisi dei DBMS spazierà dalle modalità di gestione dei dati e della sincronizzazione server-PDA, alla gestione degli errori in fase di sincronizzazione. I DBMS presi in considerazione sono SQL Server Mobile, Oracle Database Lite, DB2 Everyplace e Sql anywhere studio. Si cercherà di descrivere come questi DBMS si installano sui dispositivi mobili e di quale software necessitano lato server per realizzare la sincronizzazione server-PDA. Si descriveranno anche le caratteristiche degli applicativi installabili lato server.

1.2 I DATABASE PER DISPOSITIVI MOBILI

I database per dispositivi mobili spesso sono versioni ridotte delle controparti per server, e prevedono solo le operazioni Sql di base, questo a causa delle risorse limitate offerte dal dispositivo portatile. In aggiunta allo spazio necessario per le tabelle dei dati, i motori di database richiedono da 125 KByte a 1 MByte di memoria, a seconda di quanto il produttore sia riuscito a rendere efficiente il codice.

Un database per dispositivi mobili è un database che può essere connesso ad una rete. Il dispositivo mobile e il server possono essere connessi tramite una connessione wireless. Molto spesso viene fatto uso di una memoria cache per memorizzare i dati e le transazioni che si usano più frequentemente, in tal modo la maggior parte dei DBMS per dispositivi mobili qual'ora si verifichi un errore di connessione reperiscono i dati dalla cache e nella successiva connessione le transazioni non iniziano da capo ma riprendono dal punto in cui si sono interrotte. La memoria cache viene gestita dal software che realizza la sincronizzazione e può risiedere sia nel PDA che nel server. Se si vuole realizzare un allineamento dati PDA-server la memoria cache risiederà nel PDA in caso contrario la memoria cache risiederà nel server.

Un database per dispositivi mobili può essere utile in queste situazioni:

- Gli utenti devono essere in grado di lavorare senza una connessione wireless in caso la connessione wireless sia scarsa o non esistente.
- Le applicazioni devono provvedere una significativa interazione server-client
- Gli utenti non richiedono l'accesso ai dati veri e propri, ma solo ai dati recentemente modificati.

L'architettura dei sistemi che gestisce i database per dispositivi mobili dovrà considerare le seguenti problematiche

- Gli utenti non lavorano in una fissa posizione geografica
- Il dispositivo palmare deve consumare poca energia, deve costare poco, e deve essere portatile
- Deve prevedere una rete wireless
- Deve prevedere le limitazioni della capacità computazionale del dispositivo palmare

In una rete composta da più palmari, i database per i dispositivi mobili replicano i dati tra loro e su un database centrale. La replica dei dati tra i database per dispositivi mobili avviene con l'intervento del database centrale, questo implica che i dati prima vengono trasmessi nel database centrale e successivamente i dati aggiornati che risiedono sul database centrale si possono propagare negli altri database per dispositivi mobili. L'attività di replica implica l'esame di un database sorgente (source) per rilevare le modifiche dovute a transazioni recenti e la successiva propagazione di tali modifiche ai database di destinazione (target). La replica deve essere asincrona, dal momento che gli utenti non dispongono di una connessione costante.

In un determinato istante su un motore database possiamo avere diverse transazioni:

- transazioni completate, le cui modifiche sono già state applicate ai dati;
- transazioni in corso, le cui modifiche non sono ancora state eseguite.

Al fine di garantire la coerenza dei dati solo le transazioni completate vengono replicate.

Inoltre, è importante che la replica avvenga in modalità seriale, per mantenere lo stesso ordine in ciascun database e prevenire le incongruenze. Un'altra considerazione relativa ai database per dispositivi mobili riguarda il modo in cui vengono risolti i conflitti nel caso di aggiornamenti multipli dello stesso record.

1.3 SCHEMA DI UNA RETE DI SINCRONIZZAZIONE SERVER-PDA

Gli host fissi(Fixed Host): Eseguono le transazioni e hanno funzione di gestione dei dati con l'aiuto dei database residenti su server

Le unità mobili(MU): Sono dispositivi palmari che si muovono all'interno della regione coperta dalla connessione wireless

- Le unità mobili mantengono la connessione alla rete e ai database consolidati tramite la connessione alle stazioni di base (BS)
- Il ruolo delle unità mobili dipende dalla loro capacità computazionale.

Stazioni di base (BS): Sono delle installazioni radio bidirezionali locate in una posizione fissa, usate per comunicare con uno o più dispositivi mobili. Vengono usate da dispositivi palmari o da wireless router

Database (DBS): Archivio strutturato in modo tale da consentire la gestione dei dati

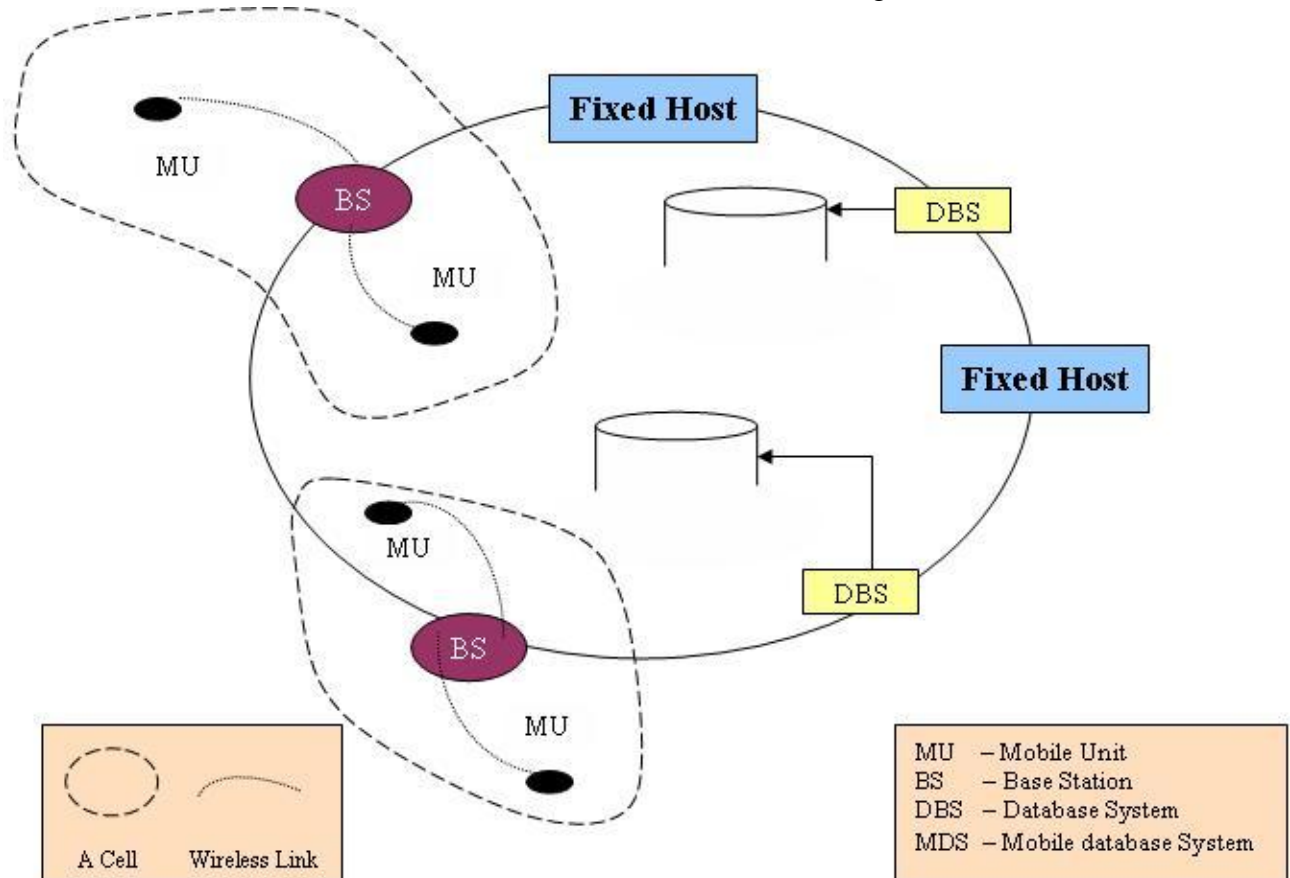


Figura 1.1: Schema di una rete di sincronizzazione server-PDA

La rete deve garantire le seguenti caratteristiche:

- Se si verifica che il dispositivo palmare lascia una cella che è gestita da un BS, si deve garantire la possibilità di effettuare transazioni e di richiedere dati dal BS che copre la nuova cella.
- I processi devono essere trasparenti e i database (DB) devono offrire gli stessi servizi a tutti i BS.

Capitolo II

SQL Server Mobile Edition

Microsoft SQL Server 2005 Mobile Edition (SQL Server Mobile), la cui versione precedente è Microsoft SQL Server 2000 Windows CE 2.0 (SQL Server CE), è un database compatto ricco di funzionalità studiato per supportare un ampio elenco di dispositivi Smart Device e Tablet PC. Gli Smart Device includono i dispositivi che eseguono Microsoft Windows CE 5.0, Microsoft Mobile 2003 per Pocket PC, Microsoft Mobile versione 5.0 per Pocket PC o Microsoft Mobile versione 5.0 per Smart Phone o successivi. L'ampio supporto di SQL Server Mobile consente agli sviluppatori di utilizzare lo stesso database su numerosi dispositivi.

Miglioramento delle funzionalità per garantire maggiore affidabilità e migliori prestazioni

SQL Server Mobile prevede molte funzionalità nuove e aggiornate che garantiscono una maggiore affidabilità e migliori prestazioni del database rispetto alla versione SQL Server CE. Questi miglioramenti riguardano tre categorie di funzionalità generali: la sincronizzazione, il motore di archiviazione e il processore delle query.

2.1 Sincronizzazione

Sono stati apportati numerosi miglioramenti al processo di replica e sincronizzazione dei dati con SQL Server Mobile, tra cui:

- Supporto di più sottoscrizioni
- Supporto multiutente e sincronizzazione
- Stato di avanzamento della sincronizzazione
- Rilevamento a livello di colonna

2.1.1 Supporto di più sottoscrizioni

Nelle versioni precedenti di SQL Server Mobile era possibile creare una sola sottoscrizione¹ per ciascun database il che significa che ad un database poteva accedere una sola applicazione predefinita. In un ambiente in cui i dati pubblicati nel database derivano da più applicazioni, lo sviluppatore doveva creare un database distinto per ciascuna applicazione e successivamente specificare il database da aprire nell'applicazione. In SQL Server Mobile un singolo database può contenere più sottoscrizioni cioè più applicazioni possono accedere ad un database; in questo modo, si riduce il carico di programmazione necessario.

¹ in inglese Subscribe e l'operazione con la quale un utente effettua la registrazione per l'accesso a un'area riservata di un sito o per la richiesta di invio di newsletter, ecc.

Il supporto di più sottoscrizioni disponibile in SQL Server Compact Edition offre i vantaggi seguenti:

1. Riduce il numero di articoli presenti in ogni pubblicazione.
2. Consente di controllare il momento di sincronizzazione dei dati provenienti da ogni pubblicazione.
3. Consente di eseguire query su più tabelle in ogni pubblicazione.

Si supponga, ad esempio, che nell'applicazione vengano visualizzate informazioni sui clienti. Metà dei dati presenti nella sottoscrizione è costituita da un elenco di clienti in sola lettura. Il resto dei dati è costituito da ordini. Gli utenti avranno la necessità di eseguire query su entrambe le tabelle, ad esempio visualizzare ordini per cliente e aggiungere nuovi ordini.

Per esigenze di prestazioni, si desidera sincronizzare i diversi tipi di dati a diverse frequenze. Per esigenze di integrità dei dati, si desidera riportare nell'elenco clienti solo le modifiche apportate nel server di pubblicazione. Queste esigenze vengono agevolmente soddisfatte dalla sottoscrizione di due diverse pubblicazioni. Ogni pubblicazione viene configurata in modo da soddisfare le esigenze di prestazioni e di integrità dei dati e il solo database di SQL Server Compact Edition conterrà le tabelle di ogni pubblicazione.

Requisiti delle sottoscrizioni multiple

Affinché un unico database contenga più sottoscrizioni è necessario applicare restrizioni:

1. Non deve esserci intersezione tra gli oggetti (tabelle, vincoli e indici) in ogni pubblicazione.
2. Non può sussistere integrità referenziale (R/I) tra le tabelle di diverse sottoscrizioni. È possibile, tuttavia, implementare l'indicizzazione per migliorare le prestazioni.
3. Le modifiche R/I sugli articoli del database di pubblicazione vengono inviate ai Sottoscrittori di SQL Server Compact Edition. Se sono presenti due sottoscrizioni all'interno di un database di SQL Server Compact Edition e il database include una tabella locale con lo stesso nome delle altre tabelle del server di pubblicazione, si verificheranno modifiche dello schema impreviste.

2.1.2 Supporto multiutente e sincronizzazione

Il supporto multiutente incluso in SQL Server Mobile consente a più applicazioni di accedere contemporaneamente allo stesso database. In particolare, consente a più utenti di un database di sincronizzare i dati senza disconnettersi dal database prima di utilizzare la replica di tipo merge o RDA (Remote Data Access).

Microsoft SQL Server 2005 Compact Edition (SQL Server Compact Edition) consente di stabilire una o più connessioni contemporanee a un database di SQL Server Compact Edition.

Consentendo a più connessioni di accedere al database di SQL Server Compact Edition si rende disponibile agli sviluppatori un modello di sviluppo semplice. Il Database Engine (Motore di database) di SQL Server Compact Edition gestisce infatti richieste provenienti da

più applicazioni oppure più richieste di connessione provenienti da una sola applicazione, quindi non è necessario che gli sviluppatori pianifichino gli accessi al database.

Se SQL Server Compact Edition consentisse l'accesso al database attraverso una sola connessione, esisterebbe un limite alla concorrenza, in quanto occorrerebbe completare una transazione prima di avviarne un'altra. Con SQL Server Compact Edition è invece possibile eseguire più applicazioni che accedono o modificano diversi set di dati contemporaneamente. Se l'accesso a una determinata risorsa del database non causa conflitto tra le applicazioni, consentire a più applicazioni di accedere al database migliora la concorrenza.

L'accesso al database di SQL Server Compact Edition può avvenire nei modi seguenti:

- Connessione singola — un'unica applicazione che richiede una singola connessione a un database di SQL Server Compact Edition.
- Connessioni multiple — un'applicazione può stabilire più connessioni a un unico database per diverse ragioni, come accedere ai dati durante la sincronizzazione.
- Applicazioni multiple — più applicazioni in esecuzione su un dispositivo possono accedere a un unico database contemporaneamente. Lo sviluppatore è sollevato dall'onere di gestire connessioni esclusive. Gli utenti possono utilizzare strumenti di database, come Query Analyzer di SQL Server Compact Edition, mentre un'altra applicazione è connessa al database.

Se una particolare applicazione esige una connessione singola e non si desidera che vi siano altre connessioni o altre applicazioni che accedono al database in contemporanea, sarà possibile aprire il database in modo esclusivo allo scopo di impedire l'accesso multiutente.

I blocchi vengono utilizzati come meccanismo di controllo degli accessi simultanei. Grazie ai blocchi, è possibile eseguire contemporaneamente più transazioni completamente isolate l'una dall'altra.

Il numero di connessioni simultanee a un database è limitato dal numero e dalla capacità delle risorse disponibili, ad esempio dalle dimensioni fisse della tabella delle sessioni, che non consente più di 256 connessioni in contemporanea.

Accesso multiutente e RDA

L'accesso multiutente in Microsoft SQL Server 2005 Compact Edition (SQL Server Compact Edition) è sempre disponibile, anche quando si utilizza RDA. È quindi possibile che nel corso di un'operazione pull o push un altro utente acceda allo stesso database di SQL Server Compact Edition.

Poiché un'operazione pull richiede che non sia presente la tabella locale specifica durante il pull e poiché la tabella viene creata durante il processo di pull dei dati dal server, non è possibile accedere alla tabella della quale viene eseguito il pull fino al termine dell'operazione stessa. Quando viene eseguito il push dei dati al server, se le modifiche ai dati vengono apportate dopo l'avvio del push, tali modifiche verranno inviate al server alla successiva operazione di push. Se i dati del server richiedono l'aggiornamento dopo una determinata operazione di push al server, l'applicazione non consente ulteriori modifiche alla

tabella fino al termine di entrambe le operazioni di push e pull. Ciò assicura che le modifiche non andranno perse al momento dell'eliminazione della tabella.

Consentendo a più connessioni di accedere al database di SQL Server Compact Edition si rende disponibile agli sviluppatori un modello di sviluppo più semplice. Il motore di database di SQL Server Compact Edition gestisce infatti richieste provenienti da più applicazioni oppure più richieste di connessione provenienti da una sola applicazione, quindi non è necessario che gli sviluppatori pianifichino gli accessi al database. Quando vengono richieste più connessioni, il motore di database di SQL Server Compact Edition blocca i dati secondo le necessità delle specifiche operazioni.

Modifica della chiave primaria durante la sincronizzazione

Per una tabella con rilevamento RDA, gli aggiornamenti, gli inserimenti e le eliminazioni vengono rilevati in base alla chiave primaria. A causa della possibilità di accesso multiutente alle colonne di chiave primaria, è importante che i valori della colonna di chiave primaria non vengano modificati durante la sincronizzazione.

La modifica di un valore nelle colonne di chiave primaria, benché rara, potrebbe causare errori e perdite di dati se si verifica durante la sincronizzazione di una tabella nella quale è stata modificata la chiave primaria.

Accesso multiutente e sincronizzazione

Poiché Microsoft SQL Server 2005 Compact Edition (SQL Server Compact Edition) supporta l'accesso multiutente, un utente può continuare a utilizzare il database mentre questo viene replicato. Possono pertanto verificarsi conflitti con le modifiche provenienti dal server di pubblicazione, ovvero conflitti di dati locali, che è necessario tenere in considerazione quando si sviluppano applicazioni che utilizzano SQL Server Compact Edition.

Effetti dell'accesso multiutente

Quando si progetta un'applicazione che utilizza SQL Server Compact Edition, è necessario tenere in considerazione gli effetti dell'accesso multiutente sul database. Nella tabella seguente vengono illustrate alcune funzionalità comuni incorporate in SQL Server Compact Edition e i problemi associati a ogni funzionalità.

Funzionalità	Problema
Blocco	<p>Durante la sincronizzazione, le modifiche provenienti dal server di pubblicazione potrebbero non venire applicate al database residente nei dispositivi mobili a causa dei blocchi dei dati. Se sul database vengono apportate modifiche a dati sui quali sono impostati blocchi di lunga durata, la sincronizzazione potrebbe non riuscire.</p> <p>Per evitare completamente il problema si deve aggiungere la</p>

	logica all'applicazione per impedire a un utente di modificare i dati durante la sincronizzazione.
Convalida	<p>Se si utilizza la convalida e si modifica il numero di righe del database di SQL Server Compact Edition durante la sincronizzazione, la convalida non riuscirà.</p> <p>Per evitare completamente il problema si deve aggiungere la logica all'applicazione per impedire a un utente di modificare i dati durante una sincronizzazione con la convalida.</p>
Reinizializzazione	<p>Durante la reinizializzazione di una sottoscrizione, tutte le tabelle utente e di sistema replicate vengono eliminate e ricreate dal livello di replica. Come avviene per le sottoscrizioni di SQL Server, le modifiche apportate dopo l'avvio della sincronizzazione vanno perse quando ha luogo la reinizializzazione.</p> <p>Per evitare questa perdita di dati, aggiungere la logica all'applicazione per impedire a un utente di modificare i dati durante la reinizializzazione.</p>
Modifiche allo schema	Tutte le operazioni DDL ¹ (modifiche allo schema) devono disporre dell'accesso esclusivo alla tabella. Una modifica allo schema non riuscirà se la tabella è utilizzata da un altro processo.
Modifiche durante la sincronizzazione	Se si verifica una modifica ai dati durante la sincronizzazione, tale modifica verrà inviata durante la sincronizzazione successiva. Se una sessione di sincronizzazione causa un conflitto locale, la riga viene risolta a livello di riga, anche se l'articolo viene rilevato a livello di colonna.

Rilevamento e risoluzione di conflitti

Quando si lavora in un ambiente multiutente, durante la sincronizzazione possono verificarsi modifiche che provocano conflitti. SQL Server Compact Edition consente di rilevare i conflitti lato client, ma non di gestire la risoluzione dei conflitti. Le informazioni sui conflitti vengono invece passate al server di pubblicazione per essere risolte durante la sincronizzazione successiva. Il programma di interfaccia che viene installato nel dispositivo SQL Analyzer è fatto di 4 finestre dove l'ultima e l'interfaccia visualizza gli errori (Notes nella seguente figura).

¹ DDL (Data Definition Language) è il linguaggio di definizione dei database utilizzato per definire le strutture logiche dei dati (campi, tabelle, etc) e la creazione di regole o di Query. Il linguaggio di definizione dei dati più utilizzato è SQL.

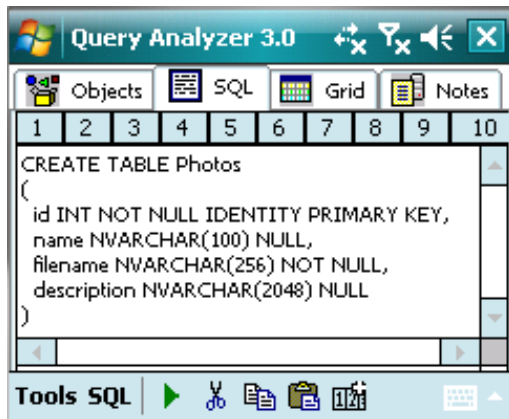


Figura 1. Query Analyzer 3.0

La maggior parte dei conflitti viene risolta nel server di pubblicazione durante la sincronizzazione successiva. Tuttavia, se si verifica un conflitto di integrità referenziale, il server di pubblicazione richiederà una risincronizzazione automatica sul dispositivo. In tal caso, non verranno eseguite più di due risincronizzazioni.

Utilizzo della proprietà `SubscriberConflicts`

Le modifiche apportate al database locale durante la sincronizzazione potrebbero provocare un conflitto locale. Se una riga del server di pubblicazione non può essere applicata nel Sottoscrittore, viene rilevato un conflitto nel Sottoscrittore e viene impostata la proprietà **`SubscriberConflicts`**. Con Sottoscrittori di SQL Server, i conflitti vengono risolti nel Sottoscrittore stesso. Tuttavia, in SQL Server Compact Edition non è disponibile un riconciliatore. Tutti i conflitti devono quindi essere risolti nel server di pubblicazione. Quando si sviluppa un'applicazione, è possibile progettare in modo da esaminare la proprietà **`SubscriberConflicts`** dopo ogni sincronizzazione. Se è impostata su un valore diverso da zero, è necessario risincronizzare i dati in modo che i conflitti possano essere risolti dal server di pubblicazione.

2.1.3 Stato di avanzamento della sincronizzazione

Gli sviluppatori possono utilizzare le interfacce API gestite per ottenere informazioni sullo stato della sincronizzazione da utilizzare per informare gli utenti.

Sincronizzazione dei dati (SQL Server Compact Edition)

La sincronizzazione dei dati garantisce che i dati nel database di Microsoft SQL Server 2005 Compact Edition (SQL Server Compact Edition) residente nel PDA corrispondano a quelli nella relativa pubblicazione. Quando si sincronizza il database di SQL Server Compact Edition, le modifiche apportate vengono inviate alla pubblicazione. SQL Server unisce quindi le modifiche con le altre modifiche apportate al database della pubblicazione, risolve gli eventuali conflitti e infine restituisce le modifiche alla sottoscrizione di SQL Server Compact Edition.

Modalità di sincronizzazione dei dati

Quando un Sottoscrittore di SQL Server Compact Edition si sincronizza con SQL Server, tutte le modifiche apportate ai dati vengono recuperate dal database della pubblicazione. Tuttavia, quando un Sottoscrittore di SQL Server Compact Edition viene sincronizzato per la prima volta, è possibile recuperare i dati direttamente dal database della pubblicazione o dal file snapshot, ovvero da un file con estensione bcp nella cartella snapshot.

Quando si sincronizzano i dati con SQL Server Compact Edition, è possibile eseguire la sincronizzazione dei dati sincrona o asincrona. In un'applicazione in cui si utilizza la sincronizzazione dei dati sincrona, si attende il completamento della sincronizzazione prima di continuare. Tuttavia, se si crea un'applicazione che utilizzi la sincronizzazione dei dati asincrona, l'applicazione potrà continuare a elaborare altre informazioni in attesa del completamento della sincronizzazione.

Avvio della sincronizzazione dei dati

I dati vengono sincronizzati quando la sincronizzazione viene richiesta dall'applicazione mediante la chiamata al metodo **Synchronize** dell'oggetto **Replication**. La sincronizzazione può inoltre venire eseguita automaticamente se si verifica un problema con una determinata sincronizzazione. Se si verifica uno degli scenari seguenti, verrà inviato un messaggio relativo alla risincronizzazione automatica dal server di pubblicazione:

1. Viene apportata una modifica allo schema sul server di pubblicazione.
2. Si utilizza un sistema di risoluzione dei conflitti sempre a favore del Sottoscrittore e non è possibile applicare i dati del Sottoscrittore nel server.
3. Si verifica un problema di integrità referenziale quando i dati del server di pubblicazione vengono applicati nel Sottoscrittore.
4. Si verifica un conflitto a livello di colonna sul server di pubblicazione. Se si verifica questo problema, il server richiede che l'intera riga venga nuovamente inviata dal Sottoscrittore.
5. Microsoft Internet Information Services viene riavviato durante la sincronizzazione.

Sincronizzazione di dati asincroni

La sincronizzazione dei dati asincroni consente a un'applicazione SQL Server 2005 Compact Edition (SQL Server Compact Edition) di continuare a elaborare le altre informazioni in attesa del completamento della sincronizzazione dei dati.

Si supponga, ad esempio, di eseguire regolarmente la sincronizzazione dei dati dell'applicazione e di inviare una grande quantità di dati tramite una connessione lenta. Durante la sincronizzazione dei dati, l'applicazione non risponderà all'input dell'utente, il quale potrebbe credere che l'applicazione non stia funzionando correttamente. Per rendere l'applicazione più semplice da usare durante la sincronizzazione dei dati, è possibile visualizzare un indicatore di stato sullo smart device che mostri lo stato della sincronizzazione. Utilizzando la sincronizzazione dei dati asincroni, è possibile avviare la sincronizzazione dei dati e quindi intercettare gli eventi di sincronizzazione non appena si verificano. Questi eventi potranno quindi essere visualizzati all'utente nell'applicazione.

Utilizzo dei metodi di sincronizzazione dei dati asincroni

Se si sta sviluppando l'applicazione SQL Server Compact Edition utilizzando Microsoft .NET Compact Framework e uno dei linguaggi di programmazione gestita (Microsoft Visual Basic o Microsoft Visual C#), le chiamate asincrone vengono gestite come le altre chiamate asincrone in Microsoft .NET Framework. Se si sta sviluppando un'applicazione nativa tramite Microsoft Visual C++ for Devices, si utilizza la proprietà **SyncStatusReport** dell'interfaccia ISSCEMerge.

Utilizzo dell'API nativa

Come per la sincronizzazione dei dati sincroni, per eseguire la sincronizzazione dei dati asincroni si utilizza l'interfaccia ISSCEMerge. Diversamente dal processo nel codice gestito, la sincronizzazione dei dati asincroni è un processo a thread singolo. Le notifiche relative alla sincronizzazione si trovano nello stesso thread della sincronizzazione. Per recuperare i rapporti sullo stato della sincronizzazione, si utilizza la proprietà **SyncStatusReport** dell'interfaccia ISSCEMerge.

2.1.4 Rilevamento a livello di colonna

Nelle precedenti versioni di SQL Server Mobile un'intera riga veniva sincronizzata con Microsoft SQL Server, anche se era stata modificata solo una colonna della riga. I tempi di sincronizzazione risultavano quindi più lunghi se le colonne non modificate della riga contenevano molti dati. In SQL Server Mobile venivano sincronizzate solo le colonne modificate.

2.2 Motore di archiviazione

Le nuove funzionalità migliorate del motore di archiviazione in SQL Server Mobile contribuiscono ad incrementare l'affidabilità e le prestazioni delle applicazioni mobili. Queste funzionalità includono:

- Motore di archiviazione modificato
- Supporto multiutente
- Riutilizzo automatico di pagine vuote

2.2.1 Motore di archiviazione modificato

L'architettura del motore di archiviazione è stata riscritta e ottimizzata per un'architettura per dispositivi mobili in cui ciascuna applicazione condivide un pool di memoria comune. L'affidabilità dei dati è ora garantita tramite il supporto effettivo di atomicità, coerenza, isolamento e durata, e in presenza di problemi di alimentazione e di connettività.

2.2.2 Supporto multiutente

Per supportare l'accesso contemporaneo di più applicazioni allo stesso database, in SQL Server Mobile sono disponibili funzionalità di blocco a livello di riga delle pagine di dati, di

blocco a livello di pagina, nonché livelli di isolamento che contribuiscono a garantire l'integrità dei dati durante la concorrenza.

Blocco (SQL Server Compact Edition)

Quando più utenti o applicazioni accedono agli stessi dati nello stesso momento, il blocco impedisce la modifica simultanea di dati. I blocchi vengono gestiti internamente da Microsoft SQL Server 2005 Compact Edition (SQL Server Compact Edition) Database Engine (Motore di database), e vengono acquisiti e rilasciati automaticamente in base alle operazioni eseguite dall'utente.

Se il blocco non viene utilizzato e più utenti aggiornano contemporaneamente gli stessi dati, i dati all'interno del database possono diventare non validi a livello logico. In tal caso, le query eseguite su questi dati potrebbero produrre risultati imprevisti.

Il blocco non viene mai richiesto direttamente dalle applicazioni. Quando inizia una transazione e i comandi vengono eseguiti nel linguaggio di query, nel linguaggio di manipolazione dei dati (DML¹, Data Manipulation Language), o nel linguaggio di definizione dei dati (DDL², Data Definition Language) all'interno della transazione, SQL Server Compact Edition blocca tutte le risorse necessarie per la protezione delle risorse al livello di isolamento desiderato.

SQL Server Compact Edition determina automaticamente quando concedere e rilasciare i blocchi per tutte le risorse. Per impostazione predefinita, il blocco a livello di riga viene utilizzato per le pagine di dati e il blocco a livello di pagina viene utilizzato per le pagine di indice. Per preservare le risorse di sistema, Gestione blocchi esegue automaticamente l'escalation dei blocchi quando viene superata una soglia configurabile sul numero di blocchi di riga. Il numero massimo di blocchi che è possibile allocare per sessione in Gestione blocchi è 262143.

È possibile modificare la granularità del blocco per alcune operazioni. In questo caso è consigliabile prestare la massima attenzione, perché la modifica della granularità del blocco può avere effetti indesiderati sulla concorrenza.

SQL Server Compact Edition consente di bloccare le risorse seguenti:

- Righe
- Pagine di indice o dati di 4 KB
- Schemi di tabella
- Tabelle
- Database

¹ DML (Data Manipulation Language): comandi SQL per l'elaborazione dei dati prelevati in tabelle di database grazie ai comandi DQL

² DDL (Data Definition Language) è il linguaggio di definizione dei database utilizzato per definire le strutture logiche dei dati (campi, tabelle, etc) e la creazione di regole o di Query. Il linguaggio di definizione dei dati più utilizzato è SQL.

Un blocco di granularità minore, ad esempio a livello di riga, aumenta la concorrenza ma presenta maggiore overhead¹ perché al blocco di un numero maggiore di righe corrisponde un numero maggiore di blocchi. Un blocco di granularità maggiore, ad esempio a livello di tabella, riduce la concorrenza perché il blocco di un'intera tabella limita l'accesso di altre transazioni alle varie sezioni della tabella. Il blocco a questo livello, tuttavia, presenta un overhead inferiore per il minor numero di blocchi.

Escalation dei blocchi (Motore di database)

L'escalation dei blocchi è il processo di conversione di un numero elevato di blocchi specifici in un numero inferiore di blocchi generici, in modo da diminuire l'overhead di sistema aumentando, contemporaneamente, la probabilità di contese di concorrenza.

Soglie di escalation dei blocchi

L'escalation dei blocchi viene avviata in presenza di una di queste condizioni:

- Una sola istruzione di Transact-SQL acquisisce almeno 5.000 blocchi su una sola tabella o indice.
- Il numero di blocchi in un'istanza di Motore di database supera la soglia di memoria o di configurazione.

Se l'escalation dei blocchi non è possibile per la presenza di conflitti di blocchi, il Motore di database attiva periodicamente l'escalation dei blocchi ogni 1.250 nuovi blocchi acquisiti.

2.2.3 Riutilizzo automatico di pagine vuote

In SQL Server Mobile è supportata la funzionalità di compattazione automatica, che consente di recuperare automaticamente le pagine di dati non utilizzate e ottimizzare lo spazio nella memoria del dispositivo.

2.3 Processore delle query

I miglioramenti apportati al processore delle query di SQL Server Mobile includono:

2.3.1 Ottimizzazione basata sui costi

Il processore delle query sfrutta i vantaggi offerti dal supporto delle statistiche nel motore di archiviazione al fine di creare piani di esecuzione delle query basati sui costi significativamente più convenienti.

¹ appesantimento, è il tempo assorbito dal sistema operativo per la gestione delle operazioni di multiprogrammazione oltre all'esecuzione dei programmi stessi.

2.3.2 Suggerimenti per i piani di esecuzione e per le query

Con SQL Server Mobile gli sviluppatori possono visualizzare il piano di esecuzione delle query e ridefinire le query in base alle proprie esigenze. È ad esempio possibile ridefinire il piano di esecuzione delle query a seconda del progetto del database, del tempo di accesso ai supporti o della velocità della CPU per specifici dispositivi supportati.

2.3.3 Nuovo oggetto SqlCeResultSet

Il nuovo oggetto **SqlCeResultSet** visualizza un cursore scorrevole e aggiornabile che permette agli sviluppatori di accedere direttamente a un database di SQL Server Mobile senza dover eseguire il doppio buffer dei dati nel dispositivo con un oggetto DataSet. Questa classe offre significativi miglioramenti nelle prestazioni rispetto all'utilizzo dell'oggetto **DataSet** e riduce la quantità di codice necessario per eseguire la stessa attività. Supporta inoltre le interfacce di associazione dati di Windows Form e può essere associata a controlli di interfaccia utente quali DataGrid, TextBox e ListBox.

2.4 Integrazione con altri prodotti Microsoft

- SQL Server Mobile è stato integrato in due applicazioni desktop: SQL Server 2005 e Microsoft Visual Studio 2005. Questa integrazione garantisce una maggior flessibilità durante la creazione e il test di database di SQL Server Mobile.

2.4.1 Integrazione con SQL Server 2005

SQL Server Management Studio permette di lavorare anche con database SQL Server Mobile. Quest'ultimo è un piccolo motore di database che gira su device Pocket PC e Windows Mobile, le cui capacità sono ovviamente ridotte rispetto al classico SQL Server. E' comunque un'ottima soluzione per avere un database relazionale.

I database SQL Server Mobile sono semplici files *.sdf*. Tramite SQL Server Management Studio possiamo aprire e lavorare questi files, indipendentemente dal fatto che essi si trovino sul nostro PC fisico, su un palmare o addirittura su un emulatore di palmare.

Apertura di un database su un PC fisico

Si può creare un file *.sdf* direttamente sul PC di sviluppo. Con SQL Server Management Studio si può aprire il file dalla finestra Connect To Server e specificare il file con il suo path locale. Si deve selezionare la voce SQL Server Mobile nella casella Server Type.

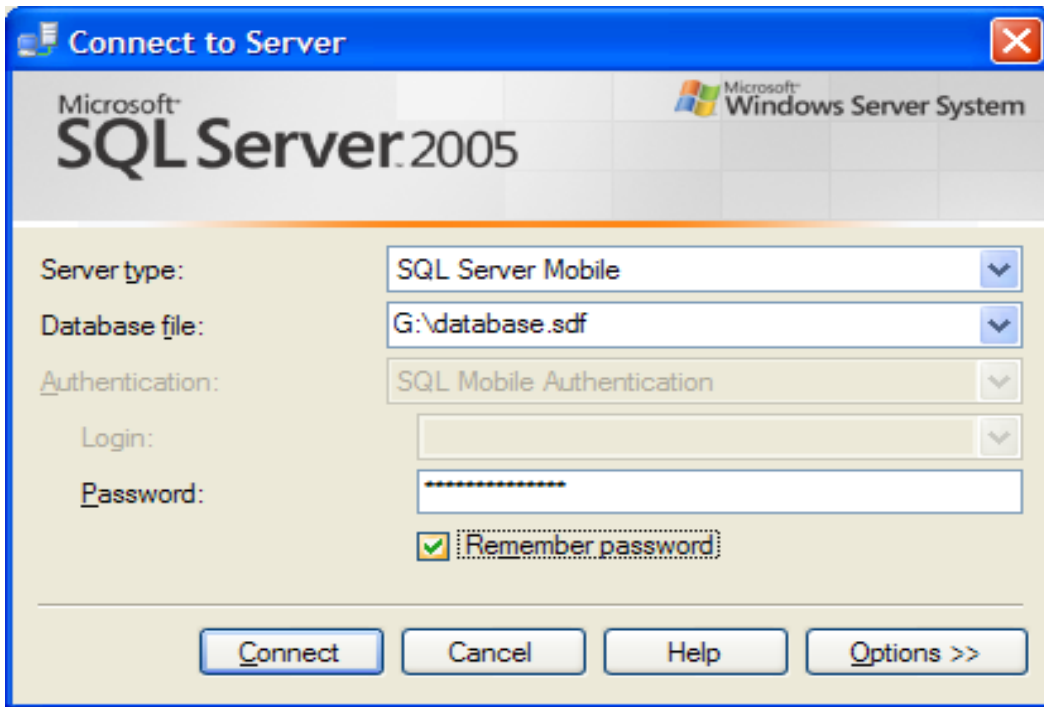


Figura 2. Apertura di un database su un PC fisico

Apertura di un database su Pocket PC/Windows Mobile

Per aprire un file *.sdf* presente su un PDA con SQL Server Management Studio è necessario che quest'ultimo sia connesso via ActiveSync oppure via Windows Mobile Device Center (questo dipende dall'OS che usate: Windows XP oppure Windows Vista). Dalla finestra Connect To Server è necessario specificare il path che punti innanzitutto al device e poi via via specificare il percorso corretto:

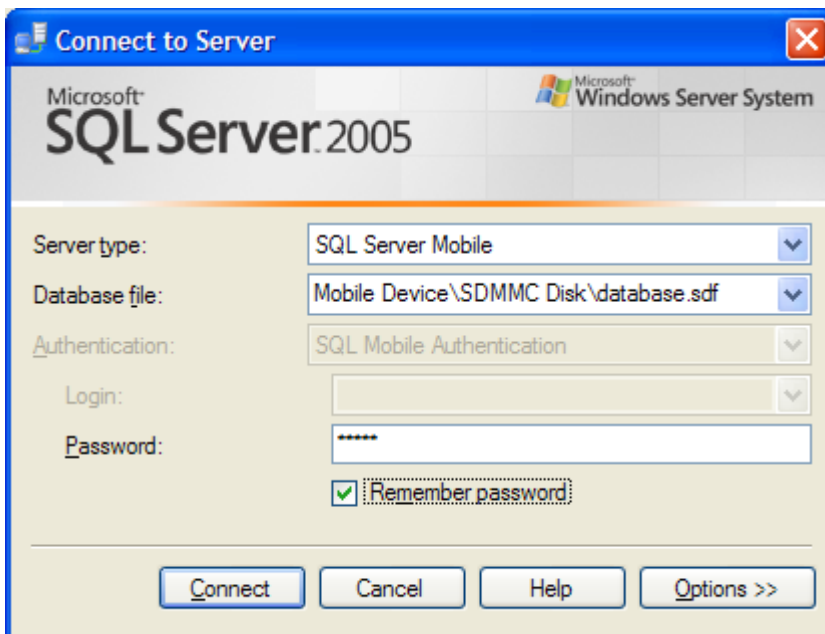


Figura 3. Apertura di un database su un PDA

Nell'immagine qui sopra si vede che il path è *Mobile Device\SDMMC Disk\database.sdf*. Se non si conosce il percorso giusto basta aprire la casella e cliccare su <Browse for more...>. Nel caso preso come esempio, il file *.sdf* è stato salvato sulla SD interna del palmare.

I vantaggi che derivano dall'utilizzo di Management Studio includono:

- Piani grafici di esecuzione delle query

I piani grafici di esecuzione delle query consentono agli sviluppatori di prendere rapidamente visione dei piani e di anticipare potenziali problemi di prestazioni delle query. È inoltre disponibile una funzionalità distinta, denominata Suggerimenti per le query, per controllare alcuni aspetti del piano di esecuzione.

- Creazione guidata nuova sottoscrizione

La Creazione guidata nuova sottoscrizione richiede un intervento minimo da parte dell'utente nelle fasi di creazione e successiva sincronizzazione di un database.

- Versione migliorata della procedura guidata Configurazione sincronizzazione Web

In SQL Server Mobile sono supportate versioni migliorate della procedura guidata Configurazione sincronizzazione Web. I miglioramenti apportati alla Procedura guidata Configurazione sincronizzazione Web includono utilizzo più semplice e supporto per le directory virtuali HTTPS.

- Integrazione con SQL Server Integration Services (SSIS)

SSIS consente agli sviluppatori di trasferire dati tra un database di SQL Server Mobile e numerosi tipi di origini dati, inclusi i database di Microsoft Access, Oracle e IBM. Include un adattatore di destinazione e uno strumenti di gestione delle connessioni per SQL Server Mobile.

- Editor SQL

Il componente Editor SQL di Management Studio è lo strumento fondamentale per progettare e testare in modo interattivo istruzioni, query e script Transact-SQL. L'Editor SQL dispone di numerose funzionalità che semplificano il processo di scrittura e di modifica delle query e del codice. Consente inoltre di modificare script creati da file o da Esplora oggetti.

- Distribuzione dei database

Dall'interno di Management Studio è possibile creare database di SQL Server Mobile su un computer locale. Questi database possono essere configurati, popolati con dati e successivamente distribuiti a più dispositivi in modo da ridurre significativamente i tempi di sviluppo e distribuzione.

Alcuni dei miglioramenti apportati in SQL Server 2005 per la replica di tipo merge offrono i seguenti vantaggi per le applicazioni di SQL Server Mobile:

- Accesso tramite interfaccia API gestita per la sincronizzazione della regola business

SQL Server 2005 presenta un'interfaccia API gestita che consente agli sviluppatori di modificare i dati della regola business prima che vengano memorizzati nel database SQL Server.

- Supporto degli articoli di solo download (tabelle)

Questa funzionalità consente di ridurre la quantità di metadati trasferiti durante la sincronizzazione iniziale, nonché il tempo di elaborazione in SQL Server quando gli sviluppatori eseguono sincronizzazioni successive.

- Supporto degli articoli partizionati (filtrati)

In molte applicazioni mobili i dati vengono filtrati per impedire agli utenti di accedere agli stessi dati da dispositivi diversi. Grazie al supporto degli articoli partizionati, gli sviluppatori possono ridurre in modo significativo i tempi di sincronizzazione e aumentare la scalabilità. Questa funzionalità è particolarmente utile per la distribuzione di molti dispositivi Smart Device.

2.4.2 Integrazione con Microsoft Visual Studio 2005

Sono stati apportati due importanti miglioramenti all'integrazione di SQL Server Mobile con Microsoft Visual Studio 2005:

- Gestione dei database

Gli sviluppatori possono gestire un database di SQL Server Mobile su un computer desktop o su dispositivi supportati utilizzando Visual Studio 2005. In particolare, grazie a questa funzionalità, possono trascinare una tabella di SQL Server Mobile in un controllo dell'applicazione e creare automaticamente l'associazione tra il controllo e la tabella.

- Installazione automatica di SQL Server Mobile

Quando si utilizza Visual Studio per creare un'applicazione di Microsoft .NET che utilizza SQL Server Mobile, il motore di SQL Server Mobile viene automaticamente installato nel dispositivo la prima volta che si distribuisce l'applicazione. Se si crea un'applicazione nativa con Visual Studio 2005, sarà necessario installare SQL Server Mobile manualmente. Si devono copiare i file .cab dal computer dove è installato Visual Studio per poi essere installati direttamente dal PDA. I file sono gli seguenti e si possono trovare nelle seguenti cartelle:

1) sqlce30.piattaforma.processore.cab che contiene i file sqlcese30.dll (il motore di SQLCE), sqlceqp30.dll (il query processor), sqlceme30.dll (contiene il codice binario necessario alla libreria .NET System.Data.SqlServerCe). Utilizzando SQLCE localmente, senza effettuare repliche, questo è l'unico componente necessario.

2) sqlce30.repl.piattaforma.processore.cab contiene tutte le dll necessarie per eseguire repliche e più precisamente sqlceca30.dll (è il Client Agent della replica, colui che contatta

il server agent esposto dalla virtual directory), sqlceoledb30.dll (necessario solo se si accede via OLE-DB al database locale, utile per applicazioni C++), sqcecompact30.dll (fornisce le funzioni di compattazione esposte in .NET dalla classe SqlCeEngine metodo Compact).

3) sqlce30.dev.lingua.piattaforma.processore.cab contiene ISQLW30, la mini interfaccia di amministrazione utilizzabile direttamente sul device (o emulatore) e sqlceerr.dll nelle varie lingue.

SQLCE e più in generale qualunque software nativo per Windows CE è scritto in codice binario e, visto che Windows CE può girare su diversi processori, anche i file .cab devono essere forniti per i vari processori e per le varie piattaforme: sotto x:\Program Files\Microsoft Visual Studio 8\SmartDevices\SDK\SQL Server\Mobile\v3.0 si trovano infatti due directory:

- 1) wce400 con le varie dll e cab per la versione 4.x di Windows CE
- 2) wce500 con le varie dll e cab per la versione 5.x di Windows CE

All'interno di queste directory trovare una directory per ogni processore supportato: ad esempio la cartella armv4i si riferisce al processore ARM V4i, mentre la cartella X86 contiene i file necessari per un device basato su processore X86. Tutti gli emulatori forniti con Visual Studio 2005 sono scritti per il processore ARM (al contrario degli emulatori di Visual Studio 2003 che erano basati su X86), quindi volendo provare un setup sul PDA con sistema operativo Windows Mobile 5 for Pocket PC occorre prendere i vari cab dalla directory armv4i all'interno della directory wce500.

Al lato server bisogna fare le seguenti operazioni:

Come prima cosa occorre installare sulla macchina Internet Information Service la parte server di SQLCE, ovvero una serie di componenti che prendono il nome di SQL Server Replication Components. L'installazione di Visual Studio 2005 appoggia il setup di tali componenti nella directory x:\Program Files\Microsoft Visual Studio 8\SmartDevices\SDK\Sql Server\Mobile\v3.0. All'interno di tale directory si trovano due file .msi: sql2kensp3a.msi che rappresenta il setup degli strumenti server per replicare i dati con Sql Server 2000 Service Pack 3 e sqlce30setupen.msi che rappresenta il setup per SQL 2005. In entrambi i casi, la dicitura "en" rappresenta la versione Inglese, l'installazione se si usa Windows Vista bisogna farlo dal prompt di MS-DOS accedendo con diritti di amministratore.

CAPITOLO III

SQL ANYWHERE 11

SQL Anywhere è un pacchetto esauriente che fornisce tecnologie per la gestione e lo scambio dei dati, permettendo lo sviluppo veloce di applicazioni che si interfacciano con database residenti su server, computer e dispositivi mobili.

SQL Anywhere offre:

3.1 Tecnologie per la gestione dei dati SQL Anywhere fornisce database per le imprese che sono disegnate per affrontare le problematiche che si possono trovare quando si opera in ambienti diversi — dal database server ad alte prestazioni al database che si può installare in un numero abbastanza grande di dispositivi mobili al interno della impresa.

SQL Anywhere mette a disposizione due tipi di database: SQL Anywhere Server e UltraLite

3.1.1 SQL Anywhere Server

SQL Anywhere Server offre molte funzionalità come il database mirroring¹ e server clustering, SQL e JAVA stored procedures, triggers, gestione dei dati al livello di riga, gestione automatica degli eventi, backup e recupero dati automatico e molte altre. SQL Anywhere Server può sopportare centinaia di utenti concorrenti e può gestire centinaia di Giga byte di dati. SQL Anywhere Server occupa poco risorse e molte delle sue particolarità che automatizzano la amministrazione fanno che l'SQL Anywhere Server sia un database ideale per installarla nei server e nelle applicazioni desktop che sono poi ampiamente usate dalle applicazioni clienti e remoti

3.1.2 UltraLite

Per ambienti che richiedono applicazioni che gestiscono quantità significativamente minori di dati si usa il database UltraLite. UltraLite è un RDBMS completo progettato specificatamente per minimizzare i requisiti di memoria e di sistema in modo da ottimizzarne l'installazione su palmari e altri dispositivi mobili (tra cui anche il BlackBerry). UltraLite fornisce il trattamento delle transazioni, e della sincronizzazione, quest'ultima realizzata attraverso un programma interno al database che permette gli scambi dati con altri database.

¹ **Mirroring:** è l'operazione di duplicazione di dati, ai fini di conservare copie di sicurezza, che possono essere usate in caso di guasto alle memorie di massa principali.

SCHEDA TECNICA DI ULTRALITE

Piattaforme supportate per il deployment¹

- Windows XP e Vista
- Windows Mobile 5.0 e successivo
- Palm OS 4.0 e successivo

3.2 Tecnologie per lo scambio dei dati SQL Anywhere offre diverse tecnologie per lo scambio dei dati, ottimizzate per affrontare le complessità che si trovano quando si deve procedere allo scambio dati attraverso reti wireless² o wired³ poco affidabili.

SQL Anywhere offre una vasta possibilità di scelta per lo scambio dati tra un sistema di impresa esistente e dispositivi mobili:

3.2.1 MobiLink—Sincronizzazione MobiLink è una applicazione inclusa in SQL Anywhere e offre sincronizzazione basata su sessioni, bidirezionale. Mobilink è un strumento ideale per scambiare dati tra un database centrale e molti database remoti UltraLite o SQL Anywhere, oppure tra una sorgente centrale di dati non relazionali e molti database UltraLite o SQL Anywhere.

Durante la sincronizzazione con MobiLink, il database remoto carica le modifiche che sono state effettuate dall'ultima sincronizzazione con il server. Appena si ricevono i dati, il server aggiorna il database centrale e poi scarica gli aggiornamenti dal database centrale ai database remoti. MobiLink assicura anche l'integrità delle transazioni nel caso in cui la connessione client-server si perda e prevede meccanismi per la risoluzione dei conflitti dei dati.

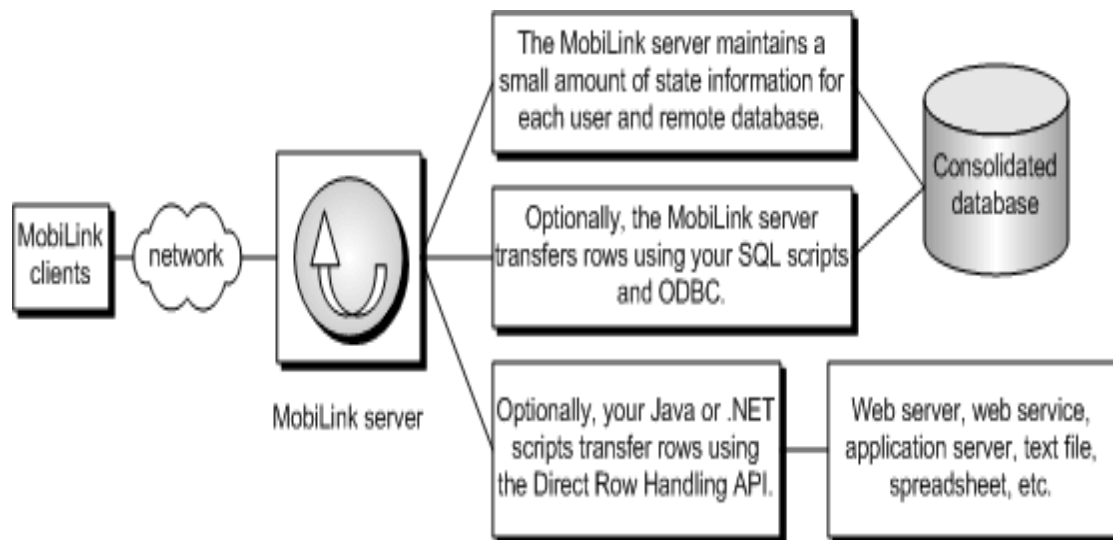


Figura 1. Sincronizzazione tramite MobiLink

¹ messa in opera, in funzione, di un'applicazione, come un database, dopo la fase di programmazione e quella di beta test.

² senza cablaggio. Sistemi collegati via etere

³ Sistemi collegati con cavi.

MobiLink permette di trasferire file alle applicazioni remote nella stessa connessione che usa per sincronizzare dati, funzionalità che è utile quando si popolano database remoti nuovi o quando si aggiorna un software.

Inoltre, MobiLink provvede la gestione diretta della riga quando si sincronizzano dati remoti con una sorgente di dati centrale. Le sorgenti di dati con i quali è possibile la sincronizzazione possono essere: una applicazione, un web server, un servizio web, una applicazione web, un file di tipo testo, un foglio di calcolo, un database non di tipo relazionale, oppure un RDBMS che non è supportato come database consolidato.

Sincronizzazione eterogenea

La tecnologia di sincronizzazione MobiLink di SQL Anywhere Studio assicura uno scambio di informazioni sicuro e bidirezionale tra Adaptive Server Anywhere remoto o il database UltraLite e numerose fonti di dati aziendali incluse Adaptive Server Anywhere, Sybase Adaptive Server Enterprise, Oracle, Microsoft SQL Server e IBM DB2. I sistemi remoti si connettono con protocolli Internet standard al server di sincronizzazione MobiLink, che comunica con il database back-end. È possibile utilizzare la tecnologia di crittografia a 128-bit per garantire dati totalmente protetti durante la trasmissione.

Protocolli specializzati, incluso il wireless

La tecnologia di sincronizzazione presente in SQL Anywhere Studio supporta la sincronizzazione wireless sulla rete LAN, nonché su quella pubblica garantendo così un accesso immediato alle informazioni aziendali. Per quanto riguarda i palmari, la sincronizzazione MobiLink supporta protocolli specializzati, tra i quali Palm Computing HotSync e Microsoft ActiveSync. MobiLink offre anche la possibilità di trasferire dati a un dispositivo remoto mediante messaggi SMS o UDP.

Introduzione agli script di sincronizzazione.

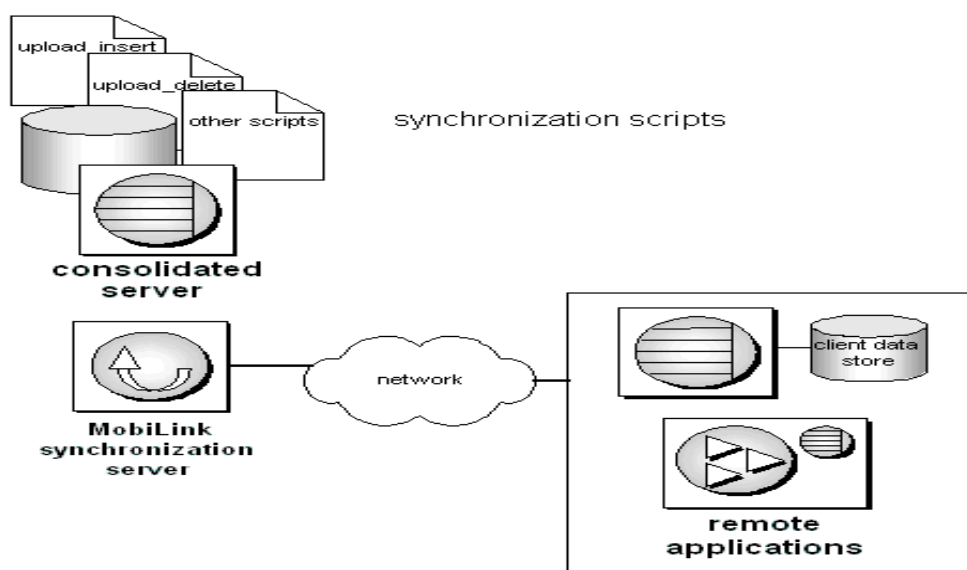


Figura 2. Sincronizzazione tramite gli script di sincronizzazione

Il processo di sincronizzazione si realizza in una serie di passi. Un specifico evento identifica ogni passo. Il processo di sincronizzazione si controlla scrivendo script associati con uno di questi eventi. Si scrive un script solo quando una specifica azione deve verificarsi in seguito ad un evento particolare. Il server MobiLink esegue ogni script quando l'evento associato si verifica. Se non si definisce un script per un particolare evento, il server MobiLink semplicemente procede al passo successivo.

Per esempio, un evento è iniziato a caricare le righe. Si può scrivere un script e associarlo a questo evento. Il server MobiLink legge questo script quando serve per la prima volta e lo esegue durante la fase di esecuzione della sincronizzazione. Se non si scrive nessun script, il server MobiLink procede immediatamente al passo successivo, che in questo caso è l'elaborazione delle righe.

Alcuni script, chiamati table script, sono associati non solo ad un evento, ma anche ad una particolare tabella nel database remoto. Il server MobiLink esegue lo script in base delle tabelle e non del database; per esempio, scaricando righe. Si possono avere diversi script associati ad un evento, ma ognuno fa riferimento a differenti tabelle.

Sincronizzazione dei dati aziendali

La sincronizzazione di SQL Anywhere Studio è stata ideata in modo che sia facile da utilizzare sia per gli amministratori del database che per gli utenti finali, permettendo così di essere sviluppata e gestita facilmente da un gran numero di utenti. L'amministrazione è notevolmente semplificata visto che tutte le funzioni amministrative vengono eseguite su un database consolidato. Garantisce affidabilità grazie all'invio sicuro dei messaggi. Se si perde la connessione alla rete oppure un messaggio viene corrotto durante la trasmissione, i dati rimangono "transazionalmente" intatti e la sincronizzazione permette di riprendere da dove si è verificata l'interruzione.

LE FUNZIONALITA' DEL MOBILINK

- Effettiva sincronizzazione bidirezionale fra i sistemi remoti e il sistema centrale che può essere: SQL Anywhere, Sybase Adaptive Server Enterprise, Oracle, Microsoft Server, IBM DB2, MySQL, sistemi ERP e Web services
- Configurazione della sincronizzazione basata su Wizard
- Supporto remoto sia ai database SQL Anywhere che a quelli UltraLite
- Supporto ai protocolli server-base di rete e di sincronizzazione multipla fra cui: TCP/IP, HTTP, HTTPS, Palm HotSync, Microsoft ActiveSync
- Criptaggio End to end
- Criptaggio ai più alti livelli a 128-bit per la comunicazione della sincronizzazione, incluso SSL/TLS usando il criptaggio RSA compatibile con i server HTTP
- Intercettazione e risoluzione programmabile dei conflitti
- Opzione di alta disponibilità per MobiLink server
- Relay server per rendere sicure le comunicazioni con gli web servers
- Ottimizzato per la sincronizzazione wireless
- Scalabile per supportare migliaia di database remoti con un singolo MobiLink server
- Supporto per il subsetting orizzontale e verticale dei dati
- Sincronizzazione basata sulla priorità e subsets multipli di dati
- File transfer
- Sincronizzazione e notifiche iniziate dal server
- Download broadcast per rendere massima l'efficienza della banda

- Sincronizzazione che si può rendere facilmente embedded¹ usando Java o Microsoft Visual Studio .NET
- Logica flessibile per l'autenticazione dell'utente
- API per applicazioni C++ e .Net applications per lanciare e monitorare l'applicazione

3.2.2 QAnywhere—scambio messaggi tra gli utenti mobili dentro l'impresa e servizi web mobili

QAnywhere offre una soluzione semplice e affidabile per scambiare messaggi tra applicazioni che risiedono in diversi dispositivi mobili. QAnywhere realizza la spedizione e ricezione dei messaggi tra le applicazioni mobili e i sistemi dell'impresa estendendo così le funzionalità offerte dal MobiLink server.

QAnywhere offre:

- Applicazioni affidabili di ricezione e spedizione di messaggi con il metodo store-and-forward.
- Comunicazioni affidabili che sono tolleranti ai difetti che si possono verificare nella rete
- Comunicazione indipendente dalla rete
- Consegna sicura dei messaggi
- Integrazione con Java Message Services—espandendo le possibilità di integrare i sistemi dell'impresa con le applicazioni basate sui database.
- Servizi Web Mobile

Funzionalità di QAnywhere

- API complete di messaggistica che forniscono un modello programmabile potente e flessibile per costruire applicazioni basate sulla messaggistica tra i dispositivi mobili
- Regole di trasmissione che ottimizzano le performance, i costi e sfruttano a pieno la banda disponibile
- Delivery dei messaggi efficiente e effettivo con capacità transazionali e di compressione
- Trasmissione e memorizzazione sicura
- Notifica push dei messaggi in attesa di essere inviati
- Connettore ai sistemi centrali back-end JMS-based
- Supporto agli Web services mobili

3.2.3 SQL Remote—replicazione SQL Remote è una applicazione inclusa in SQL Anywhere. SQL Remote è una tecnologia per replicare i dati progettata per la sincronizzazione bidirezionale tra la database consolidata e numeri abbastanza grandi di database remote, tipicamente sincronizzando molte database mobili contemporaneamente.

SQL Remote usa una architettura store-and-forward per sincronizzare i dati usando un meccanismo di trasferimento di messaggi o file come FTP o posta elettronica.

SQL Remote conserva l'integrità transazionale, diventando un strumento ideale per molte applicazioni, specialmente per le applicazioni che sono nei dispositivi mobili che operano

¹ sono i software (chiamati anche integrati) scritti non per far funzionare il computer ma apparati elettronici meno complessi (es: gli sportelli elettronici delle banche).

negli ambienti dove le connessioni non sono affidabili. I requisiti di memoria e spazio nel disco fisso sono minimi per tutti i componenti del sistema di replicazione.

FUNZIONALITA' DI SQL REMOTE

- Effettiva, sincronizzazione bidirezionale basata sui messaggi
- Sincronizzazione multipla e protocolli di rete fra cui FTP, file-based, e-mail (VIM,MAPI, SMTP)
- Intercettazione dei conflitti e risoluzione programmabile
- Supporto alla sincronizzazione wireless
- Scalabile per supportare migliaia di database remoti
- Supporto per il subsetting orizzontale e verticale dei dati
- Supporto per database SQL Anywhere

3.3 STRUMENTI DI PROGETTAZIONE E DI GESTIONE

SQL Anywhere include un pacchetto di strumenti per migliorare la progettazione e lo sviluppo di applicazioni basate sul database, altri strumenti per semplificare la gestione dei database e gli ambienti di scambio dei dati.

Gli strumenti di progettazione e di gestione inclusi in SQL Anywhere sono i seguenti:

- **Sybase Central— controllo e amministrazione centralizzato** Sybase Central è uno strumento di amministrazione e sviluppo di database integrato nel pacchetto che consente l'accesso alle impostazioni, proprietà ed eventuali opzioni attraverso un' interfaccia grafica. Sybase Central si può usare per gestire SQL Anywhere Server, UltraLite, MobiLink, QAnywhere, e altri prodotti Sybase. Per esempio MobiLink include le procedure guidate che servono per creare, sviluppare e personalizzare le applicazioni MobiLink

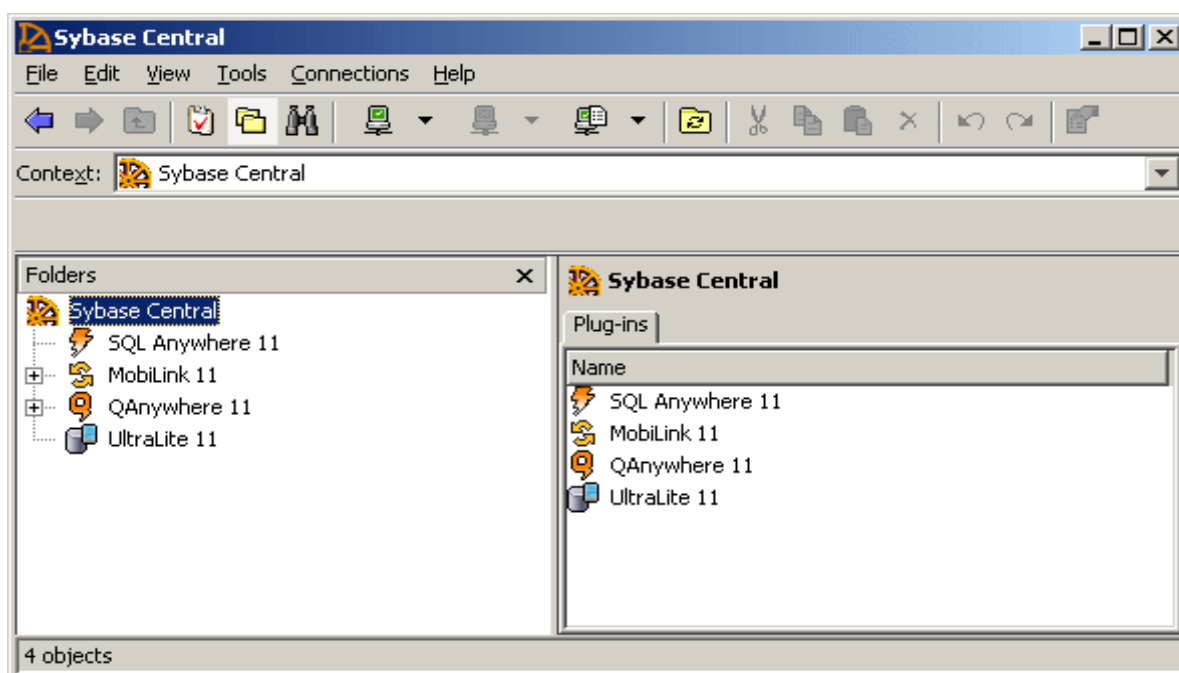


Figura 3. Sybase Central

Sybase Central offre anche visualizzazione delle statistiche delle prestazioni, le personalizzazioni delle procedure, la correzione delle stored procedure, la gestione degli

eventi, la gestione dei servizi web, e dei profili di connessione. Sybase Centrale aiuta ad amministrare molti task attraverso codice SQL inviato al database del server, oppure mediante l'uso di applicazioni di SQL Anywhere.

Un grande numero di strumenti di Sybase Central è disponibile in supporto all'analisi e monitoraggio delle prestazioni del database SQL Anywhere. Questi strumenti includono la personalizzazione delle procedure, i progetti grafici, le esecuzioni delle query, le richieste di accesso, e gli strumenti di temporizzazione.

CAPITOLO IV

DB2 EVERYPLACE

DB2 Everyplace è un DBMS relazionale e server di sincronizzazione per dispositivi mobili e distribuiti. DB2 Everyplace è la prima soluzione per dispositivi mobili in grado di riunire in un solo package¹ le funzionalità di sviluppo, distribuzione e amministrazione. DB2 Everyplace include le seguenti componenti:

- DB2 Everyplace Database - basato sugli standard di mercato, occupa pochissimo spazio di memoria 350 K ed è un vero e proprio database relazionale² che dispone una parte delle funzionalità del DB2 server.
- DB2 Everyplace Sync Server - è un servizio bi-direzionale di sincronizzazione con capacità di risoluzione dei conflitti. DB2 Everyplace Sync Server realizza lo scambio dei dati tra il database DB2 Everyplace e i database esistenti del server.
- DB2 Everyplace Personal Application Builder per PDA- Attraverso la sua interfaccia grafica e guidata, aiuta gli sviluppatori a realizzare applicazioni per database di facile utilizzo per dispositivi mobili, senza necessità di scrivere una sola linea di codice.

Il Database DB2 Everyplace

Il database DB2 Everyplace è progettata specificatamente per le applicazioni che operano nei ambienti progettati per i dispositivi mobili. Il motore database DB2 Everyplace mette a disposizione le operazioni principali relazionali ed SQL, come il Join, Group By, Order By, il Primary ed Foreign Key anche per più di una colonna, la possibilità di formulare funzioni, funzioni di Aggregazione e Vincoli. Il sistema avanzato di indicizzazione fornisce un aiuto notevole alle prestazioni per i query per le tabelle DB2 Everyplace di dimensioni medie e grandi.

DB2 Everyplace include solo le funzionalità indispensabili per i database su i dispositivi mobili così si cerca di massimizzare l'efficienza del RDBMS usando poche risorse. Funzioni per database consolidati come le subquery, la creazione dei trigger, stored procedure e funzioni definite dall'utente non sono incluse.

¹ pacchetto (di programmi, per una specifica applicazione), contenitore. Con questo termine viene indicato, anche, un piccolo disegno (icona) che rappresenta un oggetto (vedi) incorporato o collegato. Cliccando sul package, verrà richiamata l'applicazione utilizzata per creare l'oggetto, permettendone, ad esempio, la modifica o la riproduzione (file multimediale).

² insieme di database collegati fra di loro tramite dei campi chiave.

Il motore database accetta i formati di caratteri Single Byte Character Sets (SBCS), Double Byte Character Sets (DBCS) e UNICODE. Le seguenti lingue sono disponibili : Inglese, Francese, Tedesco, Italiano, Coreano, Spagnolo, Cinese tradizionale e semplificato.

In DB2 Everyplace è possibile sviluppare applicazioni database nei linguaggi C/C++ e Java™. Sono disponibili le interfacce standard DB2 Call Level Interface (CLI)¹ per le Open Database Connectivity (ODBC)² e Java Database Connectivity (JDBC)³.

Il database DB2 Everyplace è disponibile per gli sistemi operativi Palm OS, Windows CE/Pocket PC, Psion EPOC, embedded Linux, and QNX Neutrino.

DB2 Everyplace offre la possibilità di avere un database relazionale compatto nei dispositivi mobili permettendo una impresa che usa i dispositivi mobili di progettare in modo veloce e semplice applicazioni che gestiscono gli inventari e gli elenchi degli ordini. Le piccole risorse che DB2 Everyplace occupa permettono agli utenti di memorizzare molti dati. Questi dati possono essere sincronizzati usando DB2 Everyplace Sync Server.

4.1 Query-By-Example (QBE)

Figura 1. QBE visualizzazione della tabella.



¹ CLI (Command Line Interface) interfaccia della riga di comando. Interfaccia primaria per l'impostazione dei comandi di configurazione e monitoraggio del computer

² ODBC (Open DataBase Connectivity) protocollo per il collegamento verso database di varia natura.

³ JDBC (Java DataBase Connectivity) l'equivalente Java a ODBC che consente ai programmi Java di colloquiare con i database relazionali.

Query-By-Example è un'applicazione fornita con il pacchetto DB2 Everyplace per essere usata dal lato client. La figura mostra come si può usare QBE per visualizzare i dati delle tabelle in DB2 Everyplace. Per visualizzare i dati nelle tabelle si deve scorrere con i scroll bar che si vedono nel display oppure con i pulsanti su e giù del dispositivo palmare. Con DB2 Everyplace si possono visualizzare i singoli record usando il record view. Si può usare QBE per aggiornare i dati, inserire nuovi dati oppure cancellare dati dalle tabelle del database DB2 Everyplace. Gli utenti più esperti possono usare il QBE command-line processor (CLP) che è una applicazione per inserire e eseguire comandi SQL specificati dall'utente.

La Figura 2 mostra come usare il QBE command-line processor per selezionare nomi che iniziano con la lettera "L" dalla tabella con l'elenco dei numeri telefonici.

Il personale incaricato per sviluppare applicazioni nell'impresa usa QBE per creare tabelle nei dispositivi mobili e per testare le eventuali funzionalità delle applicazioni. QBE rende possibile la visualizzazione dei database nei dispositivi mobili durante lo sviluppo delle applicazioni.

Quando i dati vengono aggiornati, cancellati o inseriti vengono allineati con il database consolidato solo dopo che si è realizzata la sincronizzazione dei dati usando DB2 Everyplace Sync Server.

Figure 2. QBE command-line processor



4.2 DB2 Everyplace Sync Server

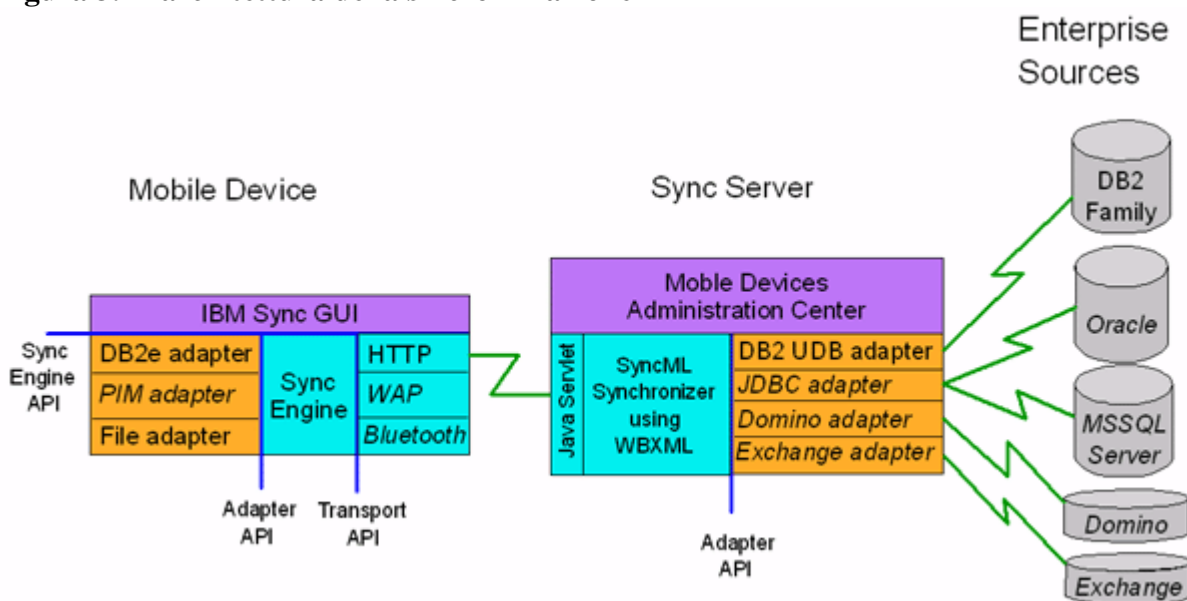
DB2 Everyplace Sync Server sincronizza dati tra il database consolidato residente sul server dell'impresa con il database DB2 Everyplace residente nei dispositivi mobili. L'architettura del Sync Server consiste nei seguenti componenti:

- IBM Sync client residente nei dispositivi mobili
- DB2 Everyplace Sync Server
- I database che sono i sorgenti dei dati

L' IBM Sync client comunica con DB2 Everyplace Sync Server e successivamente DB2 Everyplace Sync Server comunica con il database consolidato residente nel server. DB2 Everyplace Sync Server è disponibile per gli sistemi operativi Windows NT® e Everyplace Sync Server per AIX, Linux, Solaris è nella versione beta (versione non definitiva). L' IBM Sync client invece è disponibile per Palm OS e l' IBM Sync client per Windows CE/Pocket PC, Psion EPOC, embedded Linux, e QNX Neutrino attualmente si stanno testando.

La figura 3 mostra una panoramica dell' architettura del DB2 Everyplace Sync Server. Le parti colorate in blu sono le parti progettate per realizzare la sincronizzazione, una parte si trova nel server ed un'altra parte si trova nei dispositivi mobili. Le parti colorate con arancione rappresentano le parti progettate per realizzare l'interfaccia con i vari sorgenti di dati. Le parti in viola sono le parti progettate per realizzare le interfacce con l'utente e gli cilindri grigi rappresentano i sorgenti dei dati. Nella architettura sono integrati anche molti strati API¹ che danno la possibilità di avere accesso e programmare per ottenere l'estensione delle funzionalità.

Figura 3. L'architettura della sincronizzazione



La parte DB2 Everyplace Sync Server della architettura è costruita con un motore di sincronizzazione basato su SyncML. SyncML è un'iniziativa industriale sponsorizzata da molte imprese con fama internazionale come l'IBM, Lotus, Nokia, Palm, Psion, Motorola, Starfish, e Ericsson. I prodotti basati su SyncML sono progettati per essere interoperabili tra loro, usando un

¹ API (Application Programming Interface). Set di istruzioni che i programmatori usano per accedere alle funzioni del sistema operativo e ai servizi dei suoi dispositivi di periferica.

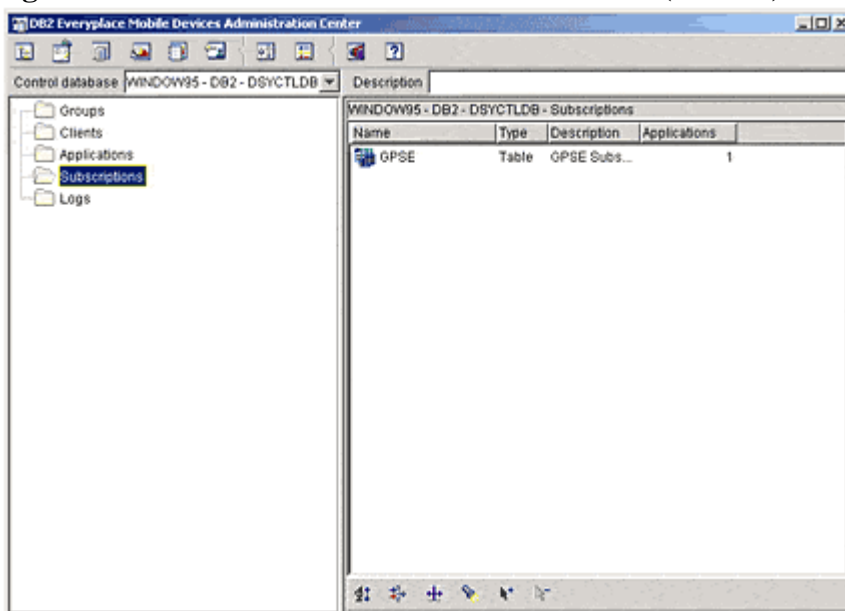
markup language¹ standardizzato e strutture di sincronizzazione per sincronizzare e replicare dati tra gli dispositivi mobili.

Il motore di sincronizzazione basato su SyncML si estende ai sorgenti dei dati inserendo strutture di interfaccia con semplici API. Altre importanti aziende possono sviluppare interfacce per i loro prodotti per avere accesso alle loro specifiche sorgenti di dati. Attualmente l'interfaccia DB2 Universal Database (UDB) è disponibile e usa tecnologia DataPropagator. Le interfacce database per la Microsoft Exchange, Lotus Domino e JDBC sono nella fase di sviluppo.

Il motore di sincronizzazione comunica per sincronizzare i dati con i clienti usando la Java servlet². Per comunicare i messaggi di sincronizzazione si usa l' WAP (Wireless Application Protocol) e WBXML (binary encoded XML). L'WBXML definisce una rappresentazione binaria compatta di XML (eXtensible Markup Language).

Il Sync Server è gestito usando l'interfaccia utente MDAC (Mobile Devices Administration Center) mostrato in figura 4. MDAC offre la possibilità di gestire una sincronizzazione per gruppi di utenti che hanno le stesse richieste per accesso ad applicazioni e dati. MDAC facilita la gestione dei gruppi di utenti, perché si possono definire dati che si devono replicare per gruppi di utenti e comunque si possono fare delle piccole modifiche se necessario alla definizione della replicazione per gli singoli utenti.

Figura 4. Mobile Devices Administration Center (MDAC)



¹ linguaggio per la descrizione di una pagina di testo (allineamento, grassetto, sottolineato etc.). Il linguaggio HTML utilizzato per la realizzazione delle pagine Web, ne è un' esempio.

² programma scritto in Java appositamente per essere eseguito da un server WEB. Questi programmi hanno la stessa funzione dei CGI script.

MDAC permette agli amministratori di avere una interfaccia amministrativa facile da usare. MDAC provvede amministrazione centralizzata per tutti gli utenti mobili, abbassando sensibilmente i costi per mantenere e gestire il sistema.

Anche IBM Sync client nei dispositivi mobili usa una architettura che offre la possibilità di avere interfacce. L' IBM Sync client è costruito basandosi su un piccolo motore di sincronizzazione che si è elaborato ed esteso usando interfacce per i dati provenienti dai PDA per accedere ai sorgenti dei dati. Attualmente interfacce client sono disponibili per i dati relazionali DB2 Everyplace e per i file. Un'interfaccia per le informazioni Personali (elenco degli indirizzi, calendari e altro) è in sviluppo. In aggiunta alle interfacce per i diversi sorgenti di dati, interfacce per diversi sistemi di trasporto si possono avere in modo da permettere la comunicazione con tipi diversi di reti. Attualmente, IBM Sync usa HTTP del protocollo TCP/IP (Transmission Control Protocol/Internet Protocol) per comunicare.

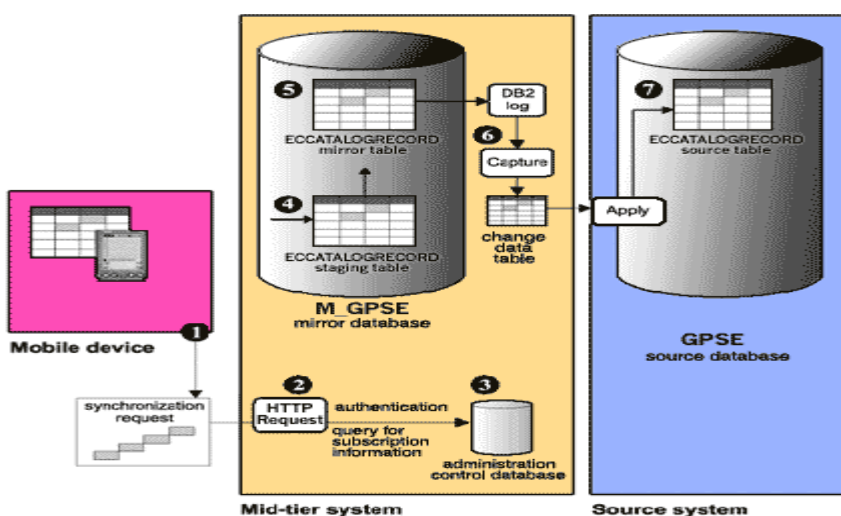
L'IBM Sync per il lato client provvede API per accedere alla funzionalità di sincronizzazione direttamente da un programma usando un specifico API (IBM Sync Engine). Le interfacce per i nuovi clienti si possono sviluppare usando l'API Client Adapter e le interfacce per usare nuovi sistemi di trasporto usando l'API Transport.

L'interfaccia utente IBM Sync fornisce menu per configurare la posizione di Sync Server, quali sorgenti di dati si vuole sincronizzare e gli username e password per accedere al server. Per sincronizzare i dati l'utente semplicemente schiaccia l'icona Sync nel display e la sincronizzazione inizia. Messaggi di stato forniscono feedback agli utenti per il progresso della sincronizzazione.

4.3 Come avviene la sincronizzazione.

La seguente figura descrive come DB2 Everyplace Sync Server sincronizza dati relazionali tra il database DB2 Everyplace e il sistema DB2 Universal Database:

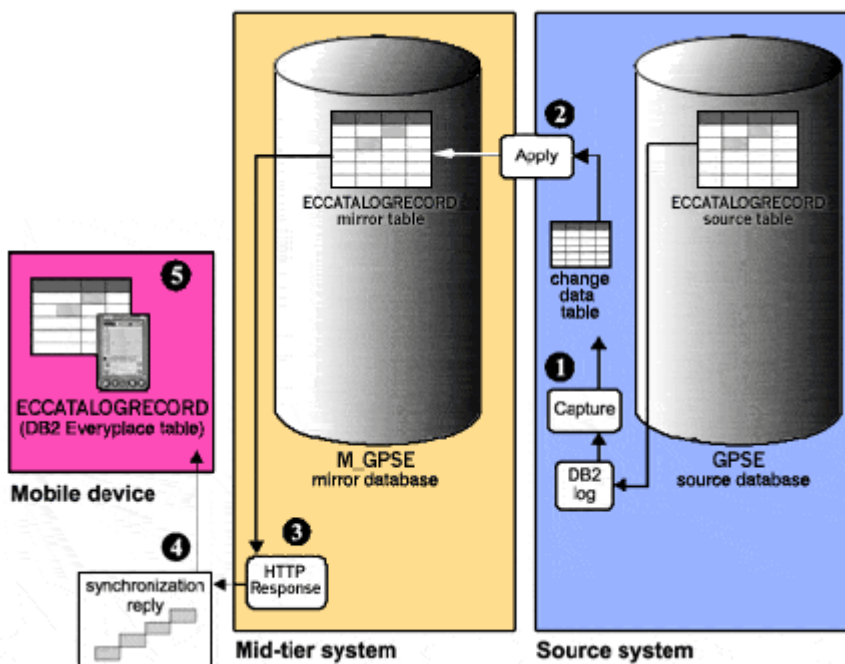
Figura 5. Sincronizzazione client - server.



Quando un utente aggiorna i dati nei dispositivi mobili, l'informazione si sincronizza durante la sessione successiva di sincronizzazione. I passi seguenti mostrano come i dati residenti nei dispositivi mobili aggiornano i dati residenti in un database consolidato. Questi passi sono rappresentati con i numeri nella figura 5:

1. L'utente inizializza la sincronizzazione schiacciando l'icona di IBM Sync. In alternativa, la sincronizzazione si può richiedere automaticamente da una applicazione per esempio usando l'API IBM Sync Engine. IBM Sync manda una richiesta di sincronizzazione al Sync Server.
2. La richiesta di sincronizzazione si identifica e poi si mette nella coda di attesa di input del Sync Server. Il software di sincronizzazione client nel dispositivo attende una risposta dal server per iniziare la sincronizzazione.
3. Dopo che la richiesta di sincronizzazione si riceve dal Sync Server nel sistema intermediario (mid-tier nella figura), l'username e il password che l'utente provvede sono identificati per verificare se è un utente Sync Server valido.
4. I dati vengono copiati in una tabella temporanea. Le tabelle temporanee aiutano a migliorare la capacità di throughput¹ del processo di richiesta di sincronizzazione perché i dati possono essere copiati mentre altri aggiornamenti possono occorrere e venire inseriti correttamente.
5. I dati vengono copiati dalla tabella temporanea alla tabella che è copia della definitiva (mirror table nella figura) M_GPSE in questo esempio, dove potenziali conflitti si risolvono. I cambiamenti che si verificano nella tabella che è copia della definitiva si registrano nel registro DB2 (DB2 log nella figura).
6. Il programma DB2 DataPropagator Capture prende i cambiamenti che si sono verificati nella tabella copia della definitiva dal registro DB2 e le scrive in una tabella dove si riportano i cambiamenti
7. Il programma DB2 DataPropagator Apply trasmette i cambiamenti dalla tabella dove si riportano i cambiamenti alla tabella definitiva, ECCATALOG.

Figure 6. Sincronizzazione Server - client.



¹ Sinonimo di: capacità di elaborazione, rendimento funzionale, velocità effettiva, rendimento di un sistema di elaborazione, cadenza di trasferimento dei dati, ecc..

I cambiamenti che si verificano nel database consolidato si trasferiscono nei dispositivi mobili durante la sincronizzazione. I passi seguenti mostrano come i dati residenti in un database consolidato aggiornano gli dati residenti nei dispositivi mobili. Questi passi sono rappresentati con i numeri nella figura 6:

1. Il programma DB2 DataPropagator Capture che è continuamente in esecuzione nel sistema che gestisce il database consolidato, cattura gli aggiornamenti che si sono verificate alla tabella definitiva dal registro DB2 (DB2 log) e gli scrive nella tabella dove si riportano i cambiamenti.
2. Il programma DB2 DataPropagator Apply applica i cambiamenti dalla tabella dove si riportano i cambiamenti alla tabella copia della definitiva, M_ECCATALOG. I conflitti che si possono verificare durante il processo si identificano e si risolvono in questo passo del processo.
3. I cambiamenti che si sono verificati ai dati si riportano alla coda di attesa nel sistema intermediario come un messaggio di risposta del processo di sincronizzazione lato server.
4. Il software che gestisce la sincronizzazione al lato client prende il messaggio di risposta del processo di sincronizzazione lato server dalla coda di attesa di output.
5. I cambiamenti nei dati si applicano alla tabella residente nel lato client e la sessione di sincronizzazione finisce.

Le funzioni DB2 Everyplace che riguardano la sicurezza.

La sicurezza è importante per proteggere informazioni confidenziali quando avviene la sincronizzazione client-server. DB2 Everyplace permette di definire gruppi di utenti che hanno diritti di accedere applicazioni, ai file e ai dati. Il username e il password si verificano durante la sincronizzazione. I dati residenti nel database consolidato del server sono protetti dagli utenti che usano dispositivi mobili impostando privilegi specifici per gli utenti che inseriscono/aggiornano/cancellano dati dal database. DB2 Everyplace usa HTTP per trasportare dati durante la sincronizzazione, quindi si può usare un firewall oppure si può definire una porta speciale per realizzare la sincronizzazione in modo sicuro.

Nel futuro, DB2 Everyplace provvederà Secure Sockets Layer (SSL)¹, autenticazione tramite certificati e codificazione dei dati locali. SSL esegue anche l'autenticazione al lato server usando una verifica del server così l'informazione non si spedisce ad un server incognito.

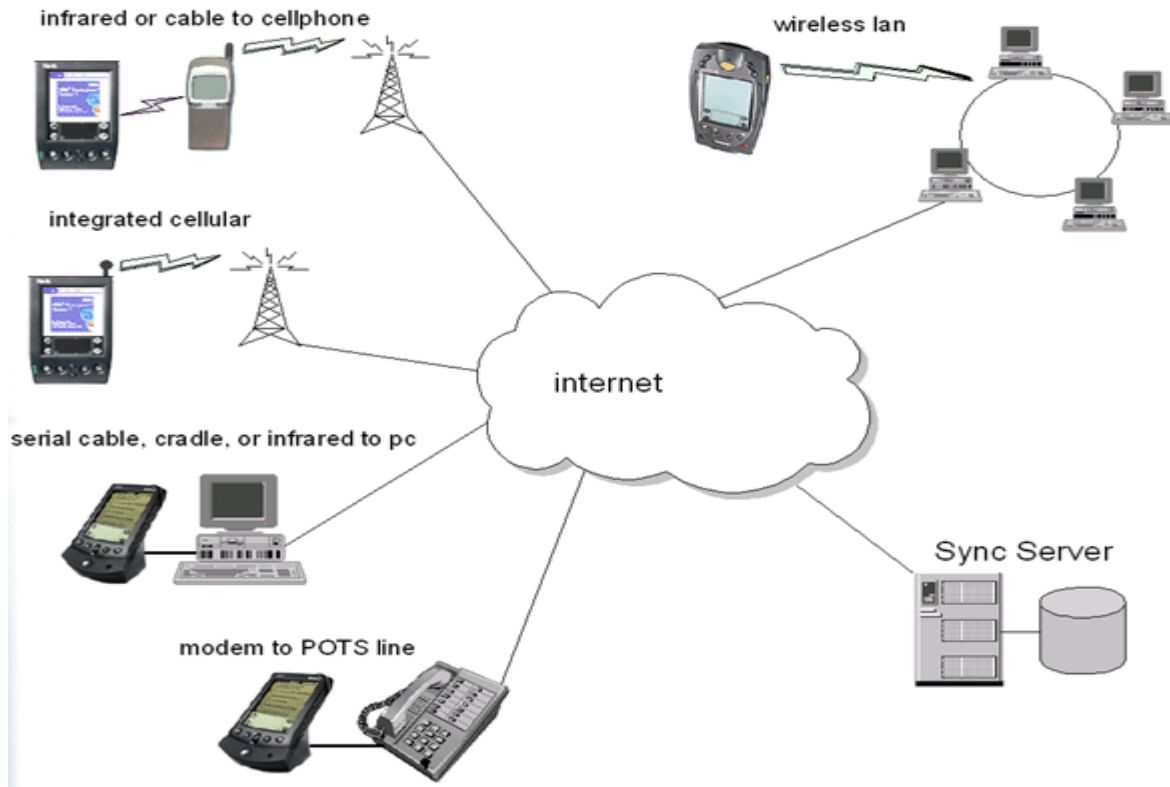
L'amministratore del server fornisce i diritti per accedere a specifici dati, predefiniti da lui, agli utenti che usano i dispositivi mobili. Non esiste nessun modo in cui si può accedere informazione per cui non si ha diritti di accesso. Gli utenti Sync Server possono accedere solo alle partizioni dei dati per cui hanno gli diritti. L'accesso si realizza usando un username ed un password per ognuno degli utenti.

Il sistema di collegamento DB2 Everyplace Sync Server

Il sistema di collegamento DB2 Everyplace Sync Server include la comunicazione wireless e wired. Esistono molte possibilità per realizzare connessioni mobili soprattutto adesso che il mercato offre molte soluzioni wireless:

¹ protocollo per la creazione di un canale di comunicazione sicuro, che impedisca l'intercettazione delle informazioni riservate come i numeri delle carte di credito.

Figura 7. Le opzioni di connessione DB2 Everyplace



Iniziando dal basso l'angolo sinistro della figura 7 e procedendo in senso orario, si possono evidenziare le seguenti opzioni di connessione:

- Il PDA che ha un modem che si connette al internet con un ISP(Internet Service Provider) tramite una linea POTS(Plain Old Telephone Service)
- Il PDA che si connette ad un PC che a sua volta è connesso ad internet tramite una connessione infrarossi, con un cradle¹ oppure con un collegamento seriale.
- Il PDA che ha una connessione integrata cellulare si connette ad internet tramite un ISP
- Il PDA che usa gli infrarossi o altro tipo di connessione per connettersi con un telefono cellulare che provvede la connessione ad internet tramite un ISP.
- Il PDA ha Wireless LAN (802.11b) si connette ad una rete wireless LAN.

Ogni protocollo di comunicazione wireless che usa TCP/IP si può usare come mezzo di propagazione dati per DB2 Everyplace perché DB2 Everyplace usa il protocollo http per trasportare dati. Gli utenti possono usare anche connessioni infrarossi oppure altre connessioni per connettersi ad un telefono e successivamente ad internet per realizzare la sincronizzazione.

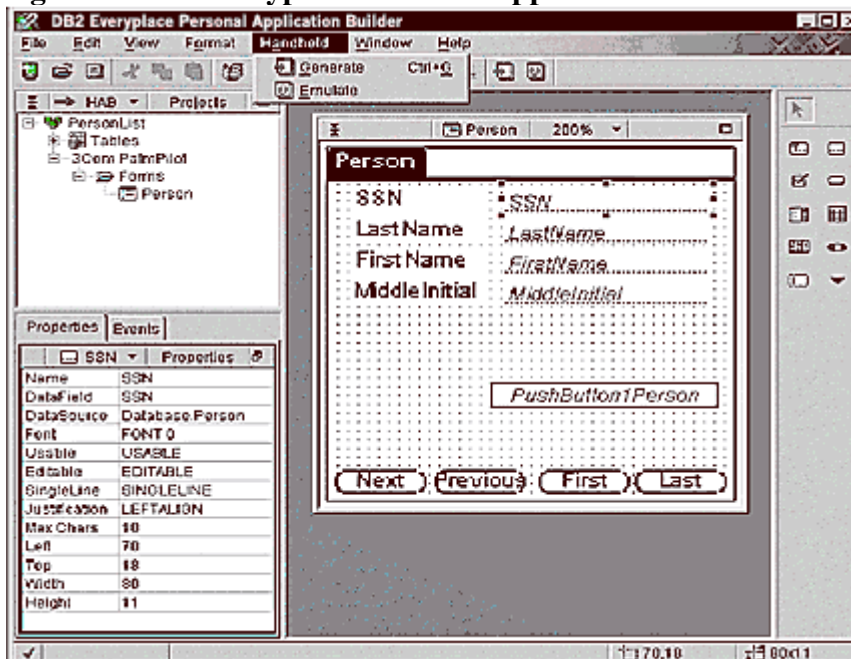
4.4 Personal Application Builder

Personal Application Builder è uno strumento che permette lo sviluppo di una applicazione in maniera facile e rapida. Con Personal Application Builder si possono creare applicazioni per gli dispositivi mobili senza scrivere una linea di codice, le applicazioni create sono progettate per

¹ così viene chiamata la base a cui si attacca un PC palmare, per collegarlo a un PC per la sincronizzazione dei dati o per caricare le batterie. Il cradle può anche essere dotato di collegamento seriale o USB.

dispositivi mobili che usano DB2 Everyplace. Attualmente Personal Application Builder offre la possibilità di integrarsi con altri strumenti per realizzare il test di diverse applicazioni.

Figure 8. DB2 Everyplace Personal Application Builder



Alcune delle tante caratteristiche incluse nel Personal Application Builder sono:

- Genera applicazioni in linguaggio C che possono accedere alle tabelle DB2 Everyplace
- Permette l'accesso al database senza scrivere del codice
- Procedure guidate per creare le tabelle del database
- Possibilità di creare prototipi che successivamente si possono copiare
- La possibilità di editare le tabelle scegliendo le opzioni da un specifico pannello
- Il supporto per gli script
- Integrazione con Palm OS e gli strumenti GNU per la compilazione, test e debugging delle applicazioni
- La possibilità di sviluppare applicazioni
- Assoluto sostegno per i database relazionali
- Non è necessario usare un interprete durante l'esecuzione delle applicazioni.

Conclusioni

DB2 Everyplace è un database relazionale robusto e compatto che offre l'opportunità di immagazzinare e manipolare dati nei dispositivi mobili. L'applicazione Query-By-Example facilita l'accesso alle tabelle DB2 Everyplace. Per venire incontro alle esigenze di un mercato internazionale DB2 Everyplace supporta molte lingue diverse e la codifica UNICODE. Una importante caratteristica di DB2 è che l'amministrazione dei processi non avviene nei dispositivi mobili perché la maggior parte dei utenti richiedono interfacce facili da usare quindi l'amministrazione dei processi avviene nel lato server. DB2 Everyplace permette lo sviluppo delle applicazioni C/C++ or Java.

DB2 Everyplace è una applicazione industriale importante. DB2 Everyplace ha le interfacce standard per scambiare dati. Le applicazioni esistenti si possono facilmente riscrivere. DB2

Everyplace supporta le API (ODBC e JDBC) standard e API personalizzati come Sync Engine API e Adapter API facilitando lo sviluppo delle applicazioni mobili per le imprese.

L'applicazione Personal Application Builder facilita la scrittura delle applicazioni. Personal Application Builder usa XML¹ e XSL² per creare l'interfaccia utente delle applicazioni per più di un PDA, non è richiesta la presenza di un interprete nel PDA e le applicazioni in genere si creano usando codice C.

DB2 Everyplace Sync Server è estensibile usando API esistenti per il server e per il client e tutto il progetto è basato sulla piattaforma SyncML. I progetti basati su SyncML promuovono l'interoperabilità tra client e server. DB2 Everyplace offre la possibilità di gestire molti utenti e nel futuro concederà la possibilità di avere anche multipli database consolidati invece di un unico database residente sul server.

¹ (eXtensible Markup Language) definisce un linguaggio per la programmazione di pagine Web. Mentre l'HTML ha dei comandi (tags) fissi questo nuovo linguaggio permette di definire i tags dando ai programmatori una versatilità maggiore nel creare pagine Web.

² (Extensible Style Language) linguaggio di programmazione per realizzare pagine Web visualizzabili su molteplici tipi di browser.

CAPITOLO V

Oracle Database Lite

Oracle Database Lite (Database Lite) è un pacchetto completo che mette a disposizione molto di più di un semplice database di piccole dimensioni. Oracle Database Lite 10g è un pacchetto con tutte le caratteristiche integrate per permettere lo sviluppo, l'inizializzazione e gestione delle applicazioni progettate per dispositivi mobili.

Le principali componenti di Oracle Database Lite sono:

1. Lo stack¹ residente nei dispositivi mobili (elemento che è diverso per ogni sistema operativo) è disponibile in tante piattaforme Windows 32-bit, Windows Mobile, Linux and Symbian OS
2. Mobile Server per la sincronizzazione, gestione e inizializzazione di applicazioni, utenti e dispositivi mobili.
3. Strumenti per progettisti che permettono di sviluppare applicazioni in modo semplice e veloce

La figura 1 mostra l'architettura tipica di un'applicazione che fa uso di Oracle Database Lite.

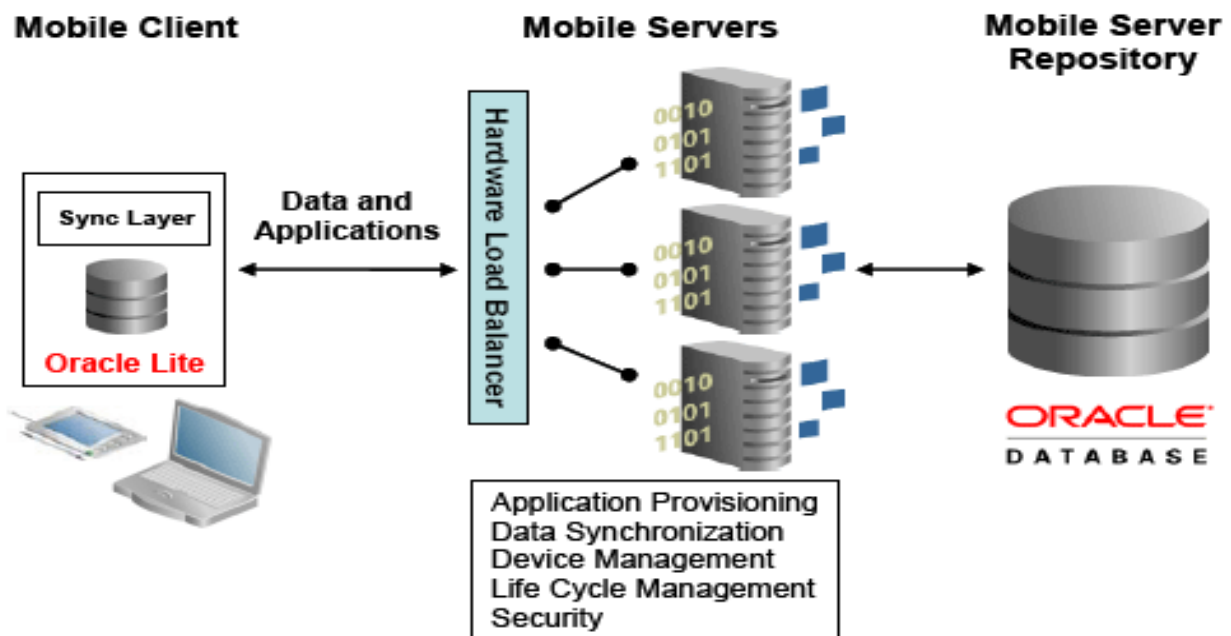


Figura 1. Architettura per il Deployment di Oracle Database Lite usando i Mobile Server

¹ area di memoria in cui i dati vengono contenuti provvisoriamente, in modo tale che l'ultimo dato inserito è il primo ad essere estratto. Lo stack viene utilizzato per l'elaborazione intermedia dei dati.

5.1 MOBILE CLIENT (La parte di Oracle Database Lite lato client)

La parte di Oracle Database Lite lato client comprende molti componenti che interagiscono per facilitare il lavoro dell'utente e per facilitare la gestione delle applicazioni per i dispositivi mobili. Il lato client comprende un database di piccole dimensioni (e gli strumenti associati) ottimizzato per dispositivi mobili. Il lato client fornisce un sistema robusto di sincronizzazione: un strumento di sincronizzazione (synchronization agent) che rende possibile la sincronizzazione automatica, una applicazione GUI¹ che permette l'invocazione manuale della sincronizzazione (mSync) e API che possono essere chiamate da un'applicazione. Lo strumento device agent permette agli amministratori di gestire i dispositivi mobili mandando comandi oppure permettendo di fare interrogazioni sui dati. Un strumento di aggiornamento permette la gestione delle applicazioni in tutta la loro fase di sviluppo permettendo di installare correttamente sui dispositivi mobili le nuove versioni dell'applicazione in maniera automatica.

L'installazione di Oracle Lite avviene attraverso un'unica installazione lato server. Il software installato include il Lite Database, mSync client, autosync client, device client, e altri strumenti che servono per gestire il database.

Quando il device client è installato, l'utente può invocare mSync client, che farà una prima sincronizzazione ed inizierà le applicazioni dell'utente, la configurazione del database e il DSN².

Il database Oracle Lite

Oracle Lite non è una versione che ha semplicemente meno funzionalità del database consolidato Oracle, è un database che viene progettato ex novo al fine di mantenere dimensioni ridotte e poter essere utilizzato da applicazioni installate sui PDA. Il database Oracle Lite è un database dove si possono depositare dati relazionali in un modo sicuro.

Le caratteristiche di Oracle Lite sono:

- Un kernel³ piccolo e flessibile
- Conformità con SQL 92 e ACID
- Java/C++/.Net Stored Procedures e Trigger
- Sincronizzazione automatica
- Amministrazione facile
- Database che occupa un massimo di memoria di 4 GB
- Offre la versione Multi-utente che supporta fino a 64 connessioni multiple.
- La possibilità di comprimere i dati
- Strumenti per caricare e scaricare i dati.

¹ (Graphical User Interface) interfaccia grafica genericamente intesa, che consente di visualizzare diversi programmi in finestre separate sullo schermo (come quella di Windows).

² (Data Source Name) nome di origine dati. Si tratta di un collegamento ad un database, residente all'interno di una rete, e accessibile solo a determinati programmi.

³ E' il nucleo di un sistema operativo, il livello più basso più vicino all' hardware, responsabile dell'intermediazione fra l'hardware ed i livelli software più alti di un sistema.

- Sistema di gestione query
- Connessioni ODBC/JDBC/ADO.Net¹
- Codifica AES² 128-bit

Oracle Lite Database si può installare nei seguenti sistemi operativi Windows 2000/2003/XP/Vista, Windows Mobile 5 & 6/CE/Pocket PC, Symbian OS, Linux e embedded Linux.

Le modalità per sincronizzare i dati tra i dispositivi mobili e il server.

Per facilitare l'accesso ai dati, gli seguenti casi devono avverarsi:

- Se gli dati residenti nel database residente nel server servono solo ad un utente, deve esistere un modo per specificare che questi dati devono arrivare solo a quel specifico utente.
- Quando un utente elabora dati fuori dalla copertura della rete, deve esistere un metodo efficiente per sincronizzare i dati dai dispositivi mobili ai server.

L'infrastruttura Oracle Database Lite è responsabile per connettere e sincronizzare applicazioni e gli dati associati a loro negli dispositivi mobili con gli rispettivi applicazioni che sono in esecuzione nel server. Oracle Database Lite offre le seguenti funzionalità:

- permette la raccolta dei dati in modo efficiente e efficace da un database residente nel server
- fornisce una infrastruttura completa progettata per usare applicazioni mobili che gestiscono database nei dispositivi mobili
- offre l'infrastruttura che rende possibile per le applicazioni, aver accesso ai dati anche quando la rete non è disponibile oppure è inaffidabile.
- è progettata per estendere le applicazioni dell' impresa e offrire dati ai dispositivi mobili.

Un' architettura implementata e progettata propriamente, con i componenti di sicurezza, aiuta a diminuire i processi che si svolgono dentro l'impresa e a ridurre i costi delle singole operazioni. Di soliti gli utenti raccolgono i dati manualmente nelle località remote, poi quando tornano in ufficio copiano i dati nel database residente nel server. Con la giusta progettazione le applicazioni mobili Oracle riducono i costi di raccolta dati, che successivamente vengono copiati attraverso l'uso di una sincronizzazione automatica verso il database consolidato Oracle nel server. Attraverso l'utilizzo delle applicazioni mobili si elimina la fase di copiatura manuale, riducendo così il rischio di errori e risparmiando nei costi di manodopera.

Si può scegliere quale metodo utilizzare tra gli seguenti per sincronizzare i dati:

- L'utente deve manualmente inizializzare ogni sincronizzazione. La sincronizzazione manuale si può inizializzare con un programma tramite una chiamata API¹ oppure l'utente inizializza un programma eseguibile per iniziare la sincronizzazione

¹ (ActiveX Data Objects) tecnologia di accesso ai dati di un database tramite oggetti ActiveX.

² (Advanced Encryption Standard) standard rientrante nelle specifiche IEEE 802.11i per la cifratura (128, 192 e 256 bit) dei dati trasmessi nelle reti wireless

- Oracle Database Lite fornisce un servizio automatico per sincronizzare dati tra il Mobile Server e l'Oracle Database Lite. Si possono specificare regole così i dati si spediscono automaticamente quando specifiche condizioni si verificano. Grazie alla possibilità di avere la sincronizzazione automatica, i dati si mantengono sempre aggiornati sia nei database residente nei dispositivi mobili sia nel database residente nel server

Oracle Database Lite fornisce una infrastruttura completa adatta a rispondere a quasi tutte le esigenze della impresa usando le seguenti componenti:

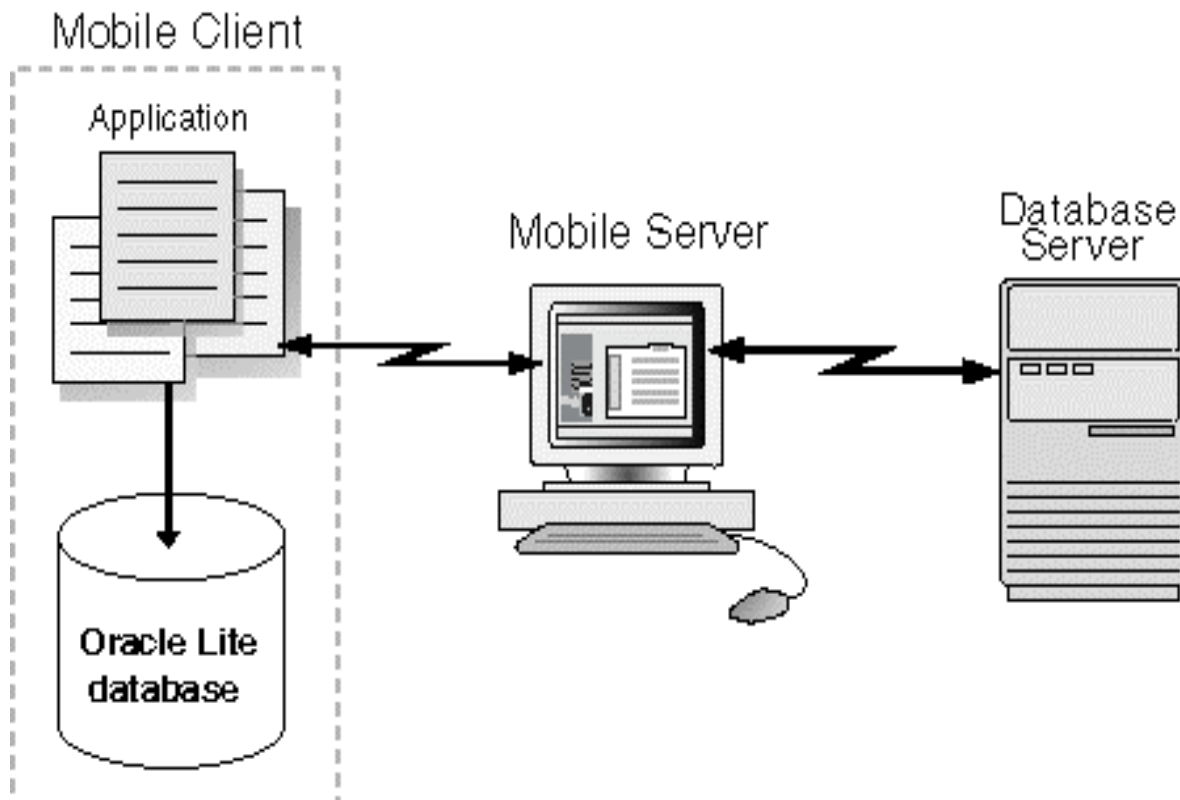


Figura 2. Sincronizzazione dati tra i dispositivi mobili e server.

- Un repository² risiede nel database consolidato e collega il database consolidato con i database che risiedono nei dispositivi mobili
- L'utente usa un database residente nel PDA chiamato il database Oracle Lite. Questi database si possono installare e gestire in quasi tutti i dispositivi da un telefono cellulare ai personal digital assistant (PDA), Table PC, Laptop, gli sistemi operativi Linux e così via.
- OC4J è una applicazione Oracle per accedere alle località remote tramite connessioni wireless e wired.

¹ (Application Programming Interface). Set di istruzioni che i programmatori usano per accedere alle funzioni del sistema operativo e ai servizi dei suoi dispositivi di periferica.

² database che contiene modelli di informazioni insieme al software eseguibile che gestisce il database.

5.2 L'ARCHITETTURA DEL ORACLE DATABASE LITE

L'architettura del database oracle lite 10g è composta da Mobile client, Mobile Server come intermediario e Mobile repository residente nel database consolidato Oracle server. Il Mobile client contiene il database Oracle Lite e le applicazioni che risiedono nei dispositivi mobili. Il Mobile Server fa le funzioni del intermediario per coordinare il processo di sincronizzazione e provvede gli strumenti di gestione per l'amministratore. Il Mobile Server repository risiede nel database consolidato Oracle ed è la che tutti gli dati provenienti da tutti gli utenti vengono immagazzinati. Il Mobile Server usa la sincronizzazione per replicare gli dati tra il database client Oracle Lite e le tabelle gestite dalle applicazioni che risiedono nel database consolidato Oracle.

Oracle Database Lite usa un modello di sincronizzazione che è personalizzato, mantiene la integrità degli dati e provvede alta sicurezza quando si sincronizza.

Un modello semplificato di come avviene la sincronizzazione è il seguente:

- Lato client – L'applicazione client comunica con il Mobile Server e trasmette gli cambiamenti fatti nel lato client al Mobile Server. Il Mobile client poi scarica qualche eventuale cambiamento per l'utente dal Mobile Server
- Lato server- Oracle Database Lite usa Mobile Server per replicare dati tra il database Mobile client e le tabelle gestite dalle applicazioni che risiedono nel database consolidato Oracle.

La descrizione di come la sincronizzazione si esegue con l'aiuto degli componenti principali di Oracle Database Lite è dimostrato nel seguente grafico

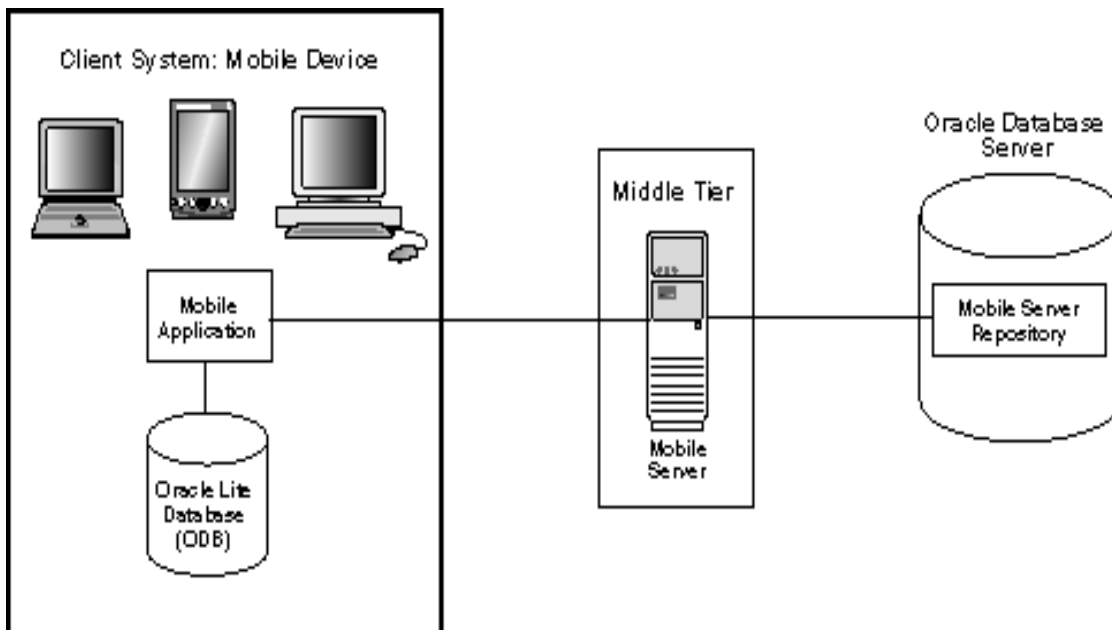


Figura 3. L'architettura del Oracle Database Lite

5.3 MOBILE SERVER

Il Mobile Server del database Oracle Lite è richiesto per fare l'intermediario per permettere agli utenti mobili di sincronizzare dati con il database dell'impresa. Il Mobile server offre anche la gestione delle applicazioni, utenti e dispositivi.

La sincronizzazione dei dati

Durante la sincronizzazione, gli cambiamenti fatti dal utente si caricano al Mobile Server per essere successivamente trasmessi al database consolidato. La sincronizzazione si può conseguire manualmente invocando l'applicazione che serve per sincronizzare ,mSync, oppure una applicazione che invoca le API che servono per sincronizzare. In particolari casi dove manca la copertura di rete oppure si richiede trasmissione senza essere intercettati (per esempio negli applicazioni militari), la sincronizzazione manuale si può realizzare usando l'opzione di sincronizzare basandosi su i File, opzione che include la generazione e il fisico trasporto degli file che si devono scambiare tra client e server.

La sincronizzazione si può anche invocare automaticamente quando al lato client le condizioni di scatto definite dal utente si avverano e al lato server si verificano le condizioni predefinite dal amministratore. Gli programmatori possono specificare regole per scattare la sincronizzazione automatica. Per esempio, si può scattare la sincronizzazione quando la quantità di cambiamenti nei dati nel lato client oppure nel lato server eccede una certa soglia oppure quando la batteria dei dispositivi mobili è sotto un livello predefinito. La sincronizzazione automatica si esegue in maniera efficiente e non richiede il blocco del database per un specifico tempo, quindi si ha la possibilità di continuare ad accedere al database locale permettendo l'utente di continuare il suo lavoro.

Il Processo di sincronizzazione

Oracle Database Lite offre un meccanismo di sincronizzazione che è asincrono che permette il supporto di tanti utenti e di soddisfare i bisogni delle imprese grandi. In un sistema di sincronizzazione sincrónico gli utenti si connettono al server per sincronizzare e rimangono connessi fino a quando tutti gli cambiamenti si applicano al database dell' impresa. Questa implementazione implicitamente limita il numero degli utenti che si possono supportare a causa degli connessioni concorrenti. Mentre un meccanismo di sincronizzazione asincrono diminuisce il tempo in cui il client è connesso al server basandosi su delle code di dati per raccogliere gli cambiamenti fatti dagli utenti per poi essere trasmessi al server e dal lato server si creano delle code di dati parallele per essere trasmesse ai dispositivi mobili.

Contemporaneamente un processo che si esegue automaticamente chiamato MGP (Message Generator and Processor) raccoglie i cambiamenti fatti da tutti gli utenti e gli copia nelle tabelle del database. MGP gestisce sia le code dei dati che entrano che quelle che escono dal server

La sincronizzazione Oracle Lite è un processo asincrono realizzato con due passi:

1. Si copiano le righe dal database residente sui dispositivi mobili nel In Queue(coda dati che entra nel server) e dal Out Queue nel database residente sui dispositivi mobili.
2. MGP(Message Generator Processor) gestisce e crea le code di dati.

La sincronizzazione asincrona Oracle Lite ha queste caratteristiche:

- Alte prestazioni
- Compressione dati
- Tecnologia scalata (supporta migliaia di utenti)
- Sincronizzazione wireless
- Rilevamento conflitti e risoluzione personalizzata
- L'uso dei checkpoint (offre la possibilità di riprendere la sincronizzazione dal checkpoint dove si era interrotta.)
- Sincronizzazione che permette accessi concorrenti (nessun utente si blocca da un altro utente)

- Architettura multi-threaded ¹
- Invocazione personalizzata della sincronizzazione
- E compatibile con i reti che usano il protocollo TCP/IP
- Integrazione di API di trasporto che permettono di usare tutte le reti wireless (esiste la possibilità di integrare anche altri tipi di API per il trasporto)

Conflitti durante la sincronizzazione

Diversi tipi di conflitti possono succedere durante la sincronizzazione. Per esempio, un utente che usa un PDA può cambiare la stessa riga che sta cambiando il server, creare righe con gli stessi valori di primary key, oppure cancellare la stessa riga che il server sta cambiando. Oracle Lite rileverà questi errori e fornisce molti meccanismi per risolverli.

Il programmatore può scegliere di usare gli meccanismi offerti da Oracle per risolvere gli errori. Questi meccanismi permettono di sovrascrivere gli cambiamenti fatti dal utente agli dati residenti nel server (l'utente domina) o vice versa (il server domina). Nel secondo caso anche se gli cambiamenti fatti nel server hanno precedenza a quelli fatti nel lato client, gli cambiamenti fatti nel lato client vengono salvati in una coda dati degli errori in modo di permettere al amministratore di rivedere e se è necessario salvare gli cambiamenti. Il programmatore può anche scegliere di risolvere gli errori della sincronizzazione in maniera personalizzata.

Errori di sincronizzazione

I cambiamenti fatti al lato client possono causare errori del tipo violazione dei vincoli di foreign key durante la sincronizzazione. Il Mobile Server non risolve automaticamente gli errori di sincronizzazione. Il Mobile Server in questi casi sposta le operazioni di transazione nella coda di errori Mobile Server. Successivamente l' amministratore del database Mobile Server può scartarli oppure adattarli e rieseguire la sincronizzazione.

La gestione del pacchetto software

Oracle Database Lite fornisce una gestione completa del pacchetto software includendo il software di installazione e di gestione dei dati e dei file negli dispositivi mobili e nei sistemi remoti. Oracle Database Lite offre la possibilità di gestire da un unico console tutti i dispositivi nella rete. Mobile Manager permette la gestione completa del lato server. Con Mobile Manager gli amministratori hanno un controllo completo e possono monitorare e gestire tutti i client mobili.

Mobile Manager include:

- La gestione delle applicazioni
- La gestione degli utenti
- La gestione del sistema
- La gestione della sincronizzazione
- La gestione degli dispositivi

La gestione delle applicazioni

Le applicazioni sono pubblicate nel Mobile Server repository dove poi vengono gestite dal strumento Mobile Manager. Il Mobile Manager permette il caricamento e la cancellazione delle applicazioni nel Mobile Server repository. Le proprietà delle applicazioni si possono modificare dal Mobile Manager. L'amministratore può anche usare Mobile Manager per connettere utenti e gruppi di utenti con le applicazioni. L'amministratore può usare Mobile Manager per sospendere/riprendere l'attività delle applicazioni e specificare i file che si possono usare da tutti.

La gestione degli utenti

¹ é la capacità di eseguire simultaneamente differenti parti di uno stesso programma

La gestione degli utenti tramite Mobile Manager permette di creare nuovi utenti e gruppi di utenti. Mobile Manager permette di gestire i gruppi specificando i diritti di accesso.

La gestione del sistema

L'interfaccia Mobile Manager permette la gestione di tutti i server. La gestione del sistema permette di ricavare informazioni dettagliate per tutte le sessioni di sincronizzazione inizializzati dagli utenti attivi.

Il file di configurazione del Mobile server, webtogo.ora, contiene i parametri operazionali di Mobile Server che servono per gestire il processo di autenticazione, i file system¹ e la sincronizzazione.

La gestione della sincronizzazione

L'interfaccia grafica Synchronization Manager permette l'amministratore di monitorare e gestire le proprietà della sincronizzazione, amministrare la configurazione, ricavare la storia della sincronizzazione e monitorare le prestazioni della MGP (Message Generator and Processor)

La gestione degli dispositivi

La gestione degli dispositivi è una richiesta cruciale per le soluzioni IT delle imprese che usano applicazioni per dispositivi mobili. Il sistema Device Management (DM) di Oracle Database Lite offre una soluzione per amministrare i dispositivi mobili e gli dati e le applicazioni remote che risiedono in questi dispositivi.

La gestione degli dispositivi ha le seguenti funzionalità:

- Ispezione remota del hardware del PDA e le proprietà del suo sistema operativo
- Ispezione remota e modifica delle proprietà delle applicazioni
- Raccoglie informazione dal database residente nel PDA e lo sincronizza con il database residente nel server.
- La gestione software degli PDA
- Modifica dei file di configurazione dei dispositivi.

5.4 IL MOBILE DEVELOPMENT KIT (MDK)

Oracle Database Lite fornisce supporto per sviluppare facilmente applicazioni mobili off-line². I programmatori possono scegliere di aggiungere il MDK dalla installazione di Oracle Database Lite e di non installare semplicemente il database e i file di Database Lite per semplificare la progettazione, inizializzazione e il debugging delle applicazioni mobili. Il MDK fornisce accesso ai API di sincronizzazione e ai API di accesso ai database. Oracle Database Lite offre la possibilità di sviluppare applicazioni nei linguaggi Java, C/C++ e .Net. Quindi gli API di sincronizzazione sono disponibili per tutti questi linguaggi. Database Lite offre anche le API per JDBC, ODBC, ADO.NET per il database Oracle Lite.

Strumenti Disponibili Per Sviluppo di applicazioni

Oracle Database Lite supporta gli seguenti IDE³ e strumenti per lo sviluppo di applicazioni:

¹ questo termine indica, in un sistema operativo, la struttura generale in cui i file vengono nominati, memorizzati e organizzati.

² fuori linea. Al contrario di Online, indica lo stato di non connessione di un sistema alla rete.

³ (Integrated Development Environment) ambiente di sviluppo integrato.

- Oracle JDeveloper 10g
- Eclipse
- IntelliJ
- Microsoft Visual Studio .Net
- Sybase Powerbuilder
- Altri IDE

Modelli per Sviluppare applicazioni

Database Lite offre ai programmatori una varietà di modelli. I programmatori possono costruire applicazioni locali, sestanti e che operano off-line usando Java, .Net o C/C++.

Oracle Database Lite offre la possibilità di sviluppare applicazioni web-based che si possono implementare usando applet oppure Java sevellet/JSP¹ però che si possono eseguire anche in assenza di una connessione. Nella modalità disconnessa, una applicazione server fornisce queste “applicazioni web” al browser² dei dispositivi mobili. L’applicazione residente sui dispositivi mobili continuerà a funzionare e apparire come se fosse connesso alla rete. Quando il cliente si riconnette al Mobile Server, il database Oracle Lite si sincronizza e l’applicazione residente nel dispositivo mobile viene alimentata dal applicazione residente al server come un qualsiasi applicazione web.

Il metodo per sviluppare applicazioni web-based risparmia sforzi di progettazione perché permette la stessa applicazione di essere usata in ufficio dove esiste copertura di rete continua e permette il funzionamento della applicazione anche nelle località remote dove la connessione può essere sporadica. Un modello web-based offre la portabilità della applicazione nelle piattaforme che si può desiderare di integrarla nel futuro. Oracle Database Lite usa OC4J servlet come server locale della applicazione.

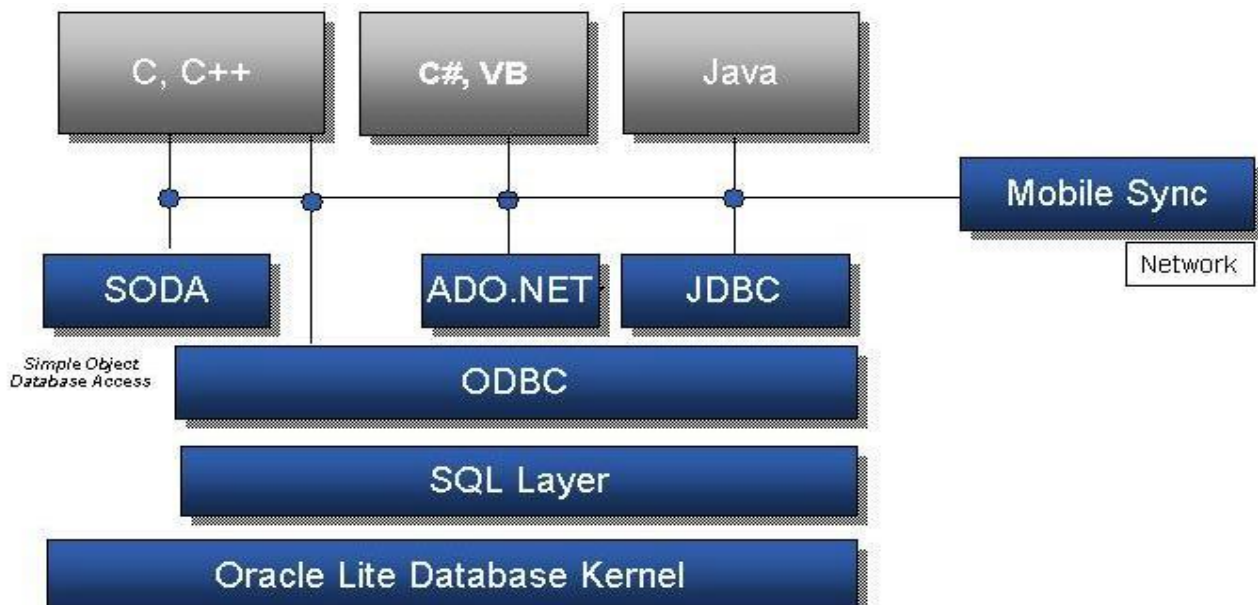


Figura 4. Modello di sviluppo Oracle Lite

¹ (Java Server Pages) sistema di scripting, simile alle ASP di Microsoft, in grado di realizzare pagine web interattive a contenuto dinamico.

² sinonimo di programma per navigare nel WWW.

Mobile Database Workbench (MDW)

Mobile Database Workbench (MDW) è uno strumento GUI per lo sviluppo delle applicazioni mobile che è installato insieme con MDK. Il Mobile Database Workbench usa una procedura guidata ben definita per guidare i programmatori delle applicazioni mobile attraverso il processo di costruzione, test e miglioramento delle loro schemi della applicazione prima di scrivere il codice della applicazione. I programmatori possono usare MDK per specificare stored procedures/triggers e dichiarazioni DDL¹ che si devono poi eseguire nel database residente sul dispositivo mobile. In più i programmatori hanno la possibilità di riusare e manipolare componenti di schemi preesistenti.

Packaging Wizard

Packaging Wizard è un strumento GUI indipendente installato insieme con MDK che permette ai programmatori di creare, modificare e pubblicare una applicazione nel Mobile Server.

Il programmatore può usare il Packaging Wizard per specificare le metadata (per esempio nome, locazione nel server mobile etc.), il tipo di file (binarie, immagine, servlet/JSP etc.) e il tipo di database client che l'applicazione usa. Il Packaging Wizard poi comprime tutte gli componenti della applicazione (file eseguibili, DLL, immagini etc.) in un file unico che serve per essere eseguito dal dispositivo mobile installando l'applicazione.

Questo file si può pubblicare nel Mobile Server usando Packaging Wizard. Dopo l'installazione dell'applicazione, l'amministratore può controllare i privilegi d'accesso degli utenti e degli gruppi.

Mobile Server Repository Diagnostic Tool (MSRDT)

Il Mobile Server Repository Diagnostic Tool si può usare per verificare l'integrità del Mobile Repository. MSRDT esegue un controllo di consistenza nel repository e provvede un rapporto dettagliato degli potenziali problemi permettendo agli amministratori di identificare le corruzioni nel repository e di intraprendere azioni correttive.

5.5 CONFIGURAZIONE DELLA INIZIALIZZAZIONE DEL APPLICAZIONE

Oracle Database Lite offre due architetture di inizializzazione principali. Le applicazioni possono sincronizzarsi direttamente con il Mobile Server (come mostrato in figura 4) e possono essere scritte per accedere ad un Oracle Lite Database .

¹ (Data Definition Language) è il linguaggio di definizione dei database utilizzato per definire le strutture logiche dei dati (campi, tabelle, etc) e la creazione di regole o di Query. Il linguaggio di definizione dei dati più utilizzato è SQL.

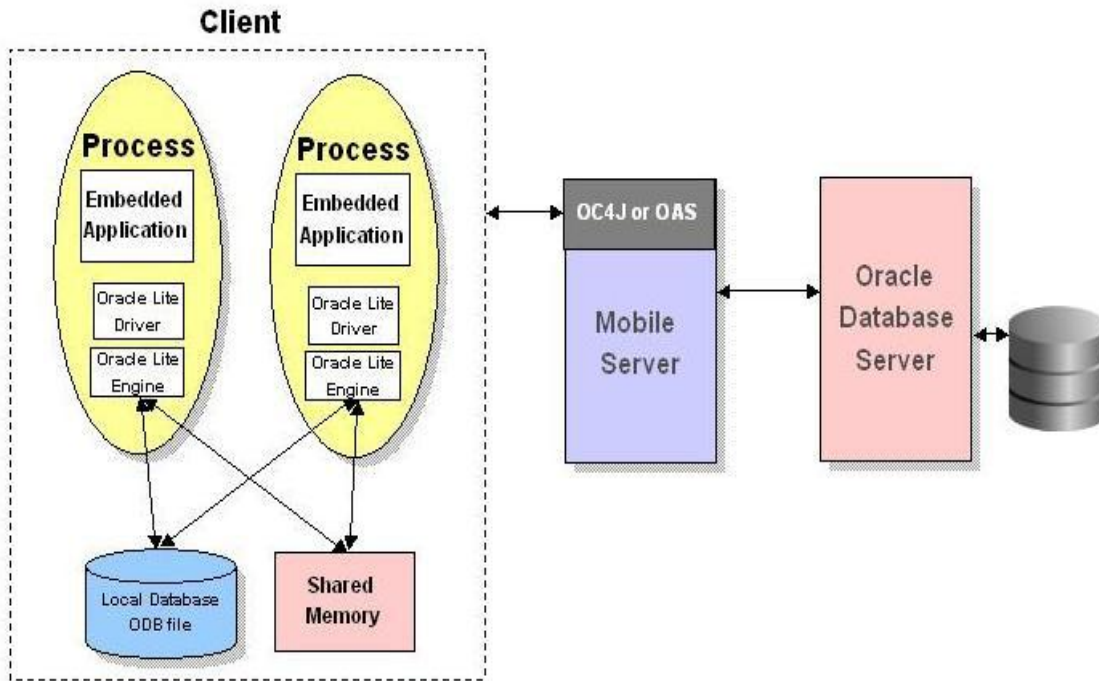


Figura 5. Sincronizzazione con il Mobile Server

In base ai bisogni della impresa si può decidere di usare anche una inizializzazione Branch Office per il software Oracle Database Lite (come mostrato in figura 5). In una configurazione Branch Office l'Oracle Lite Database provvede un mini-server che può sopportare fino a 64 clienti connessi contemporaneamente architettura realizzata con un canale condiviso. La configurazione Branch Office è usata in molti scenari implementati nei sistemi usati nelle imprese. Ogni server Branch Office si può gestire con l'interfaccia centrale usando Mobile Manager.

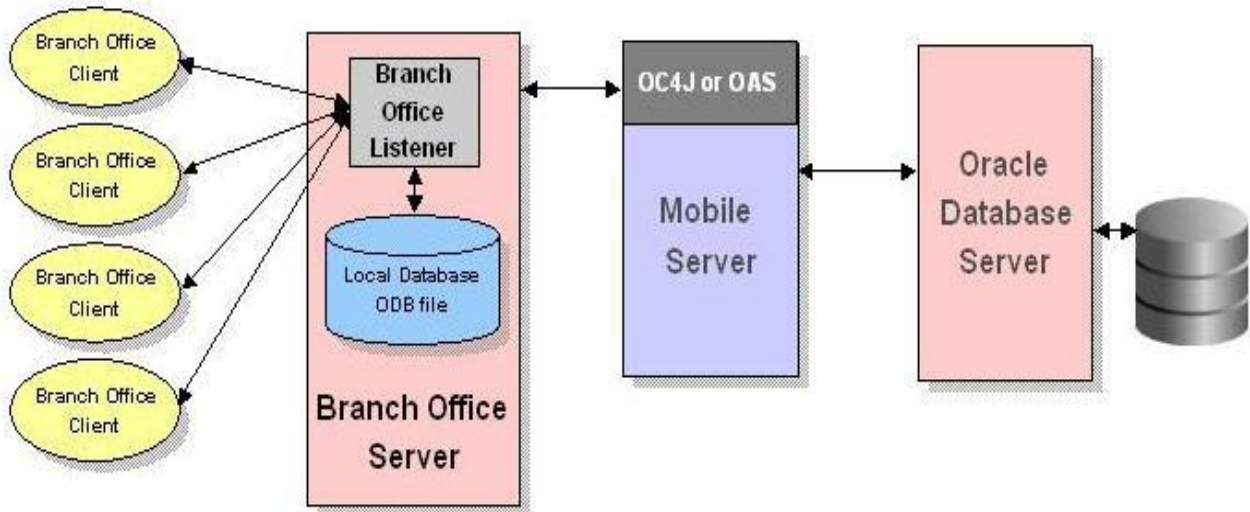


Figura 6. Configurazione Branch del Office Oracle Lite Database

CAPITOLO VI

CONFRONTO TRA I DBMS PRESI IN CONSIDERAZIONE

Nel capitolo II si è descritto il DBMS fornito da Microsoft. SQL Server Mobile è un DBMS integrato con Visual Studio oppure con SQL Server Management Studio. Si è descritto il modo in cui si installa nel palmare ed è stato l'unico DBMS che personalmente ho installato. Ho visto che SQL Server Mobile necessita il supporto di SQL Server Management Studio residente nel lato Server perché con Query Analyzer si riesce solo a creare il database. Query Analyzer non permette di creare le tabelle direttamente dal palmare perché non si può visualizzare tutta la finestra che si apre dalla procedura guidata e questa è una restrizione che ha SQL Server Mobile.

Ho provato di creare una applicazione di gestione con i strumenti offerti dalla Microsoft e ho visto che è indispensabile installare il SQL Server Management Studio per realizzare la sincronizzazione e per creare i database .sdf che servono per il palmare. Al lato server avevo il sistema operativo Windows Vista che non permetteva di installare i programmi direttamente ma bisognava installarle dal MS-DOS con diritti di amministratore. Il problema che mi ha impedito di continuare il progetto è che SQL Server Management Studio installato sotto Windows Vista non permette di scegliere sin dal inizio il server SQL Server Mobile che è la modalità che si deve scegliere per creare, gestire e realizzare la sincronizzazione con il database residente nel PDA. Quindi se si desidera creare una applicazione con i programmi offerti da Microsoft si deve per forza installare al lato server il sistema operativo Windows XP.

DB2 Everyplace offre una piattaforma che permette di usare e condividere efficientemente le applicazioni anche in lingue diverse. Questo significa che la stessa applicazione può essere per esempio usata da un utente tedesco e uno giapponese, il che permette di sviluppare applicazioni per aziende che operano in un ambiente internazionale.

Il supporto delle varie piattaforme è un fattore chiave nella scelta di un database mobile: nessuno naturalmente vuole investire tempo di formazione e sviluppo su una piattaforma che potrebbe diventare obsoleta. L'offerta di Microsoft lavora in ambiente Win32 e Windows CE. I prodotti di IBM, Oracle e Sybase supportano Linux, Palm OS, QNX Neutrino, Symbian EPOC, Windows CE e Win32.

Al momento è Sybase SQL Anywhere a dominare il mercato dei database mobili cosa che personalmente terrei in considerazione se in un futuro progetterò una applicazione per palmari. Questo prodotto vanta oltre 6 milioni di utenti, in più di 10.000 siti diversi, e ha il 68% del mercato secondo uno studio condotto di recente da Gartner Dataquest.

Tipo di DB	COSTI	Sistema operativo installato nel palmare	Ambiente di sviluppo	Programmi installati nel palmare	Sincronizzazione Client - Server	Piattaforma che usa molte lingue	Accesso multiutente al database	Gestione eventuali errori
SQL server 2005 compact edition	Distribuito con sql server oppure con visual studio	Windows CE 2.0 o superiore	Sql server 2005, Visual studio	Sql Analyzer	Procedura Merge replication e RDA(remote data access)	NO	SI	Risoluzione lato server e risincronizzazione
Oracle database lite mobile server 10g	Gratuito (bisogna iscriversi al sito)	Windows Mobile 5, Windows Ce Standard Sdk 5.0	Microsoft Visual Studio .Net, Sybase Powerbuilder		Sincronizzazione automatica con trigger definiti dal programmatore	NO	SI	Risoluzione personalizzata. Si può scegliere di sovrascrivere i cambiamenti fatti nel PDA a quelli fatti nel server o vice versa
DB2 Everyplace	72.50 \$	Windows CE 2.0 o superiore, Palm OS 3.0 e 3.1	IBM DB2	IBM DB2 Database Engine IBM Sync Query By Example (QBE)	Usando DB2 Everyplace Sync Server and DB2 Everyplace Sync Client	SI	SI	Gestione automatica dal programma.
Sql anywhere studio	Gratuito	Windows CE 3.0 o superiore.	Microsoft Visual Basic, Visual Studio .NET	Ultralite	Sybase MobiLink	NO	SI	Riprende la sincronizzazione da dove si è verificata l'interruzione

Capitolo VII

Conclusioni

In questa tesi sono stati descritti i principali DBMS ideati per i dispositivi mobili, le loro caratteristiche, il modo in cui gestiscono i dati e il modo in cui replicano i dati tra i database residenti nei PDA e nel server. Sono stati analizzati i principali applicativi per palmari e i programmi necessari lato server per gestire la sincronizzazione server-PDA.

La tesi ha descritto in maniera dettagliata SQL Server Mobile, Oracle Database Lite, DB2 Everyplace e Sql anywhere studio.

Il confronto tra i suddetti DBMS ha evidenziato come diversi fattori influenzino la scelta di un DBMS per dispositivo mobile. Tale scelta non deve essere guidata solamente da aspetti economici, ma deve essere tener conto delle diverse funzionalità fornite dai DBMS disponibili attualmente, funzionalità quali, per esempio, il multilinguismo, la compatibilità con diversi sistemi operativi (lato palmare e lato server), la sicurezza dei dati, i metodi di sincronizzazione.

Bibliografia

<http://blog.vivendobyte.net/archive/2008/11/03/681.aspx>

<http://blogs.devleap.it/rob/archive/2005/05/02/3411.aspx>

<http://blogs.devleap.com/articolidevleap/archive/2006/08/20/13266.aspx>

[http://msdn.microsoft.com/it-it/library/ayee3tzx\(VS.80\).aspx](http://msdn.microsoft.com/it-it/library/ayee3tzx(VS.80).aspx)

[http://msdn.microsoft.com/it-it/library/ms171750\(SQL.90\).aspx](http://msdn.microsoft.com/it-it/library/ms171750(SQL.90).aspx)

[http://msdn.microsoft.com/it-it/library/ms171746\(SQL.90\).aspx](http://msdn.microsoft.com/it-it/library/ms171746(SQL.90).aspx)

<http://www.sybase.be/detail?id=1021296>

documentazione online disponibile sul sito di sybase

Tesi di Daniele Carafoli disponibile sul sito di dbgroup.

<http://www.ibm.com/developerworks/library/wi-cloud/>

<http://www.ibm.com/developerworks/library/it-novcc00/>

<http://www.oracle.com/technology/products/lite/index.html>

documentazione scaricabile da questo sito

<http://www.oracle.com/technology/documentation/database10gR3.html>