

Solomon IV

SQL Server™ 7.0 White Paper

Designed for

Microsoft®
BackOffice®

Designed for

Microsoft®
Windows NT®
Windows®98

Table of Contents

Solomon IV for BackOffice – SQL Server 7.0 White Paper	1
Introduction	1
Architecture	1
Applications	2
Reporting	2
Customization	2
Database.....	2
System Database.....	2
Application Database	3
Security	3
SQL Server 7.0	4
High Performance and Scalability	4
Relational Engine	4
Query Processor.....	4
Parallel Query.....	4
Indexing	4
Storage Engine	5
Database Files.....	5
Row Level Locking	5
Dynamic Memory Management	5
Low Administrative Overhead and Cost of Ownership.....	6
Configuration Settings	6
Microsoft Management Console.....	6
Wizards.....	6
Backup and Restore	9
Data Management Objects	11
Online Analytical Processing Services	12
Data Transformation Services	12
Online Analytical Processing	12
Replication.....	12
Conclusion.....	12

Solomon IV for BackOffice – SQL Server 7.0 White Paper

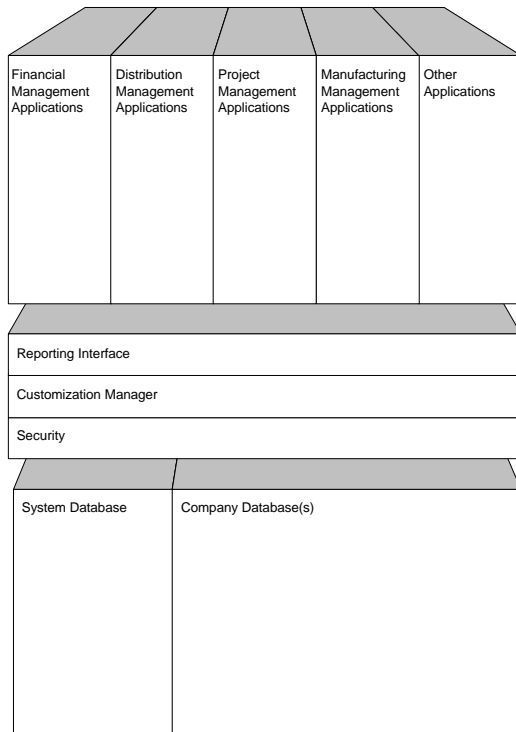
Introduction

Solomon IV for BackOffice is a full-featured suite of financial management and enterprise-wide applications optimized for Microsoft SQL Server 7.0. The integration between Solomon IV for BackOffice and SQL Server 7.0 provides customers with a powerful set of business solutions based on the market-leading business platform, Microsoft® BackOffice.

This white paper covers the advantages SQL Server 7.0 brings to Solomon IV for BackOffice. By explaining the overall Solomon IV database architecture and technical strategy, this paper also explains how Solomon IV is optimized for Microsoft SQL Server 7.0.

Architecture

Figure 1 Solomon IV for BackOffice Architecture



Solomon IV for BackOffice consists of a suite of enterprise-wide application series (Financial, Distribution, Project Management, Manufacturing, etc.)

Each of the Solomon IV applications are built with Solomon's award-winning Tools for Visual Basic®. This provides each application with consistent facilities, such as security, data entry, customization, and reporting.

Solomon's open architecture allows other applications and SQL Server utilities to easily access Solomon data (unless prevented by SQL Server security). For instance, if a user is granted the proper SQL Server security rights, Solomon's financial tables can be replicated to a separate server for offline analysis. Alternately, ODBC data sources allow applications like Access, Excel and Word can be used to insert Solomon data directly into Office documents.

Applications

The Solomon IV application components include functions that are most familiar to users, such as General Ledger, Accounts Payable, Accounts Receivable, etc. All Solomon IV applications are created in Visual Basic using Solomon IV's Tools for Visual Basic (VB Tools).

Each application is designed to take optimal advantage of the database server. For instance, data entry screens provide efficient navigation through data by using server side cursors. Additionally, much of the data intensive processing, like releasing or posting, is accomplished at the server using Transact SQL stored procedures. This gives optimum performance and reduced network traffic.

Some Solomon IV processes, such as producing reports, require the workstation to execute part of the process. The Solomon IV Application Server module offers the ability to host these processes off the client's workstation in favor of the application server.

Reporting

Solomon IV embeds two industry-leading reporting tools with its product: FRx™ financial report writer and Seagate Crystal Reports™ for Solomon IV.

Each Solomon IV application is delivered with a set of standard reports. This does not limit users to only those report definitions provided by Solomon IV. At runtime, a user can specify alternate selection or sort criteria. Each report can also be customized to offer information not provided on the base report. Additionally, users can write their own reports based on requirements specific to each site.

Customization

Each Solomon IV application is fully customizable. The Solomon IV Customization Manager module allows users to change the look, feel, and behavior of Solomon IV screens. Furthermore, the Customization Manager allows fields from custom tables to be included in a standard Solomon IV screen and allows custom stored procedures to be created to perform processing that is particular to a customization.

Database

Solomon IV for BackOffice uses a minimum of two SQL Server databases per installation. The Solomon "System" database holds site-specific information that is used for managing the installation of the product. One or more "Application" databases contain the accounting specific information.

System Database

The Solomon System database holds information like security, registration, customization definitions and application server requests. The Solomon Kernel components and System Utilities use these facilities to manage the entire suite of Solomon applications.

For the Multi-Company feature of Solomon IV for BackOffice, the system database also contains data that is used to support multi-company processing. The Solomon IV for BackOffice multi-company configuration provides for many different scenarios that could involve one or more application databases. Each scenario is managed by the System database, regardless of the number of Application databases, for optimal flexibility. The System database also provides a set of logical views to the application databases so that users need only know the company that they wish to access (rather than the names of the System or the Application database).

Application Database

A physical Application Database can contain data for one or more companies. The multi-company feature is also designed to allow data for multiple companies to reside in separate application databases, depending on the requirements of the user installation. Each table in the Application Database is defined with a clustered primary key and may contain additional indexes depending on specific requirements. With these keys and indexes, application and stored procedure queries can rapidly find the required row or rows for display or update in a screen, report, or process.

The application database also has a standard set of views on tables or groups of joined tables. A view is a “virtual table” that represents another way of accessing the data in a physical table or set of joined tables. Accessing data through these views provides consistent data sources to ensure accurate reconcilable processes and reports.

The design of the individual Solomon IV for BackOffice applications allows the processing of data to take place where it offers the best performance... either on the client workstation or on the SQL Server in the form of stored procedures. Typically, complex processes in Solomon IV that use intense calculations or have other significant data movement requirements are written as Transact-SQL stored procedures and execute entirely on the server. Where data movement is not a heavy consideration or more control is required at the workstation, the processing is written in Visual Basic.

Whether written in Visual Basic, or Transact-SQL, all database updates through Solomon IV applications maintain complete transactional integrity of the data. This means that an entire transaction is either completed in its entirety (committed) or not completed at all (rolled back). The data in Solomon IV databases is never left in an inconsistent state.

Security

Solomon IV's security model offers a functional view of security so administrators do not have to grant rights to particular database objects such as tables, views, or stored procedures. Instead, users are restricted based on the screen or function they are allowed to perform. An administrator will also want to secure data at the database level to prevent unauthorized access to Solomon data from other applications, such as Microsoft Access or Excel.

Since the multi-company feature is designed to let the data reside in a single database or separate databases, a user might be faced with the task of remembering which database holds a particular company's data. An innovative feature of Solomon IV allows users to “sign on” to a company, rather than a database. This metaphor means the user never has to remember a database name or location, only the company name. Security is provided at the company level, enabling the restriction of user access to sensitive data.

Within the context of company sign-on, Solomon also supports a unified log in feature. If the SQL Server's NT Domain has already validated a user, Solomon IV's sign-on procedure can use that verification to connect to the proper company database. Users can take advantage of this feature and never have to remember a separate password for Solomon login.

SQL Server 7.0

Version 7.0 of Microsoft SQL Server offers many improvements to the Solomon IV for BackOffice installation. These improvements are grouped into three categories: high performance and scalability, low administrative overhead and cost of ownership, and online analytical processing services.

High Performance and Scalability

Much of SQL Server has been redesigned from the ground up so that the database server offers high performance across all installations. Users can run small single user installations running the SQL Server on Windows 95/98 or NT Workstation up to enterprise installations running SQL Server on NT Clusters.

Relational Engine

The Relational Engine is that part of SQL Server that processes requests for data from the various clients. This engine has been redesigned to offer many improvements. Specific new features in the Query Processor and Indexes offer considerable advantage to Solomon IV applications.

Query Processor

The query processor now supports several new join and aggregation techniques that offer better performance for Solomon IV applications. Joins allow SQL statements to process fields from multiple tables as if they were in a single table or view. A single Solomon application database contains many tables and Solomon IV for BackOffice must use joins quite frequently. SQL Server also gathers query statistics automatically, offering up-to-date statistics for the query processor as well as reducing the need to perform the UPDATE STATISTICS command.

Parallel Query

SQL Server 7.0 can split a single query across multiple processors in a symmetric multi-processor (SMP) system. Many of Solomon IV's processes and reports that use complex queries will benefit from this feature in multiple processor NT Servers.

Indexing

SQL Server can now use multiple indexes in a single query. This offers advantages to standard Solomon processes and reports as well as customized user reports by providing fast query response without necessarily adding a new index to support the query. The flexibility of Solomon's report interface allows standard and custom reports to be sorted or filtered on many different fields. This means that new indexes need not be added to the standard schema in order to support user-specific requirements.

Each of Solomon's application tables has a clustered primary key, which provide for very fast navigation through the table. SQL Server 7.0 has improved the clustering mechanism so that changing the location of a data row (through insert, update, or delete) does not require other indexes to be rebuilt. Solomon IV benefits by improved performance in both data entry and processing operations.

Storage Engine

The Storage Engine on SQL Server handles all I/O between the query processor and the disk subsystem. SQL Server 7.0 offers a completely redesigned storage engine that eliminates database devices as well as enables row level locking.

Database Files

SQL Server 7.0 stores each of its databases on one or more native operating system files, rather than pre-allocated database devices. These files can dynamically grow or shrink, which means that an administrator no longer has to be concerned with how big a database device has to be to support a database or how much space is left on a device. Spreading the I/O load across multiple physical hard disks can spread a single database over multiple files, providing improved performance.

Row Level Locking

The SQL Server 7.0 storage engine has added row level locking as a standard feature. Row level locking causes individual rows, rather than pages, to be locked as a row is updated. This means that there will be less contention in Solomon IV environments.

For instance, when a Solomon IV invoice document is being inserted or updated, the single row on which the single invoice resides remains locked until all detail transaction lines for the invoice have also been inserted or updated. Since only the particular rows within the tables in question (rather than all data pages on which the rows reside) are being locked, this reduces the chances of multiple Solomon users attempting to update table pages at the same time and thus results in faster overall data entry.

SQL Server also supports dynamic lock escalation. This feature reduces the number of locks required by escalating the type of lock from row to page to table if enough locks are required. In normal processing like data entry, row level locking provides greater efficiency for Solomon IV applications. However, Solomon IV stored procedures that may update many rows at once benefit from lock escalation automatically. There is no need for a developer or DBA to specify which locks should be escalated.

Dynamic Memory Management

Aside from database files dynamically changing size, SQL Server also dynamically manages memory allocation, taking and giving back memory to the system as needed by the server to process requests from connected users. As users disconnect, memory is freed and given back to the operating system. This technique lets SQL Server scale down to a very small memory footprint on Mobile SQL. It also allows SQL Server to use NT Server memory resources more efficiently.

Low Administrative Overhead and Cost of Ownership

Configuration Settings

SQL Server manages all of its resources without a DBA needing to configure any settings. SQL Server allocates resources as needed for number of locks, database connections, open database objects, and many other settings that were formerly configured by settings that could not be changed while SQL Server was running.

Microsoft Management Console

The Microsoft Management Console (MMC) is the new standard management interface for SQL Server Enterprise Manager. It offers the ability to maintain several SQL Server databases from a single MMC. All database objects (tables, views, stored procedures, etc.) as well as every wizard and maintenance dialog are available from the SQL Server Enterprise Manager in MMC.

Wizards

Over 20 new wizards are available to accomplish most administrative tasks. Some of these tasks include creation of backups, databases, stored procedures, maintenance plans, alerts, and jobs.

The following example series of dialogs illustrates using the “Create Job Wizard” to build an automated job to post all unposted batches every night. To accomplish this, a set of Transact-SQL commands is scheduled to execute as part of the job.

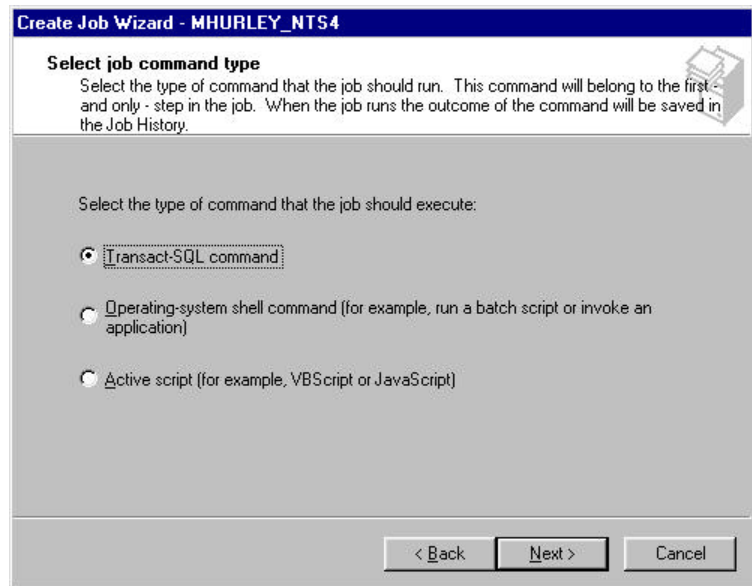


Figure 2 Create Job Wizard – Select job command type

The Transact-SQL script shown in Figure 3 executes several Solomon IV stored procedures to post all transactions that are left unposted in the current period.

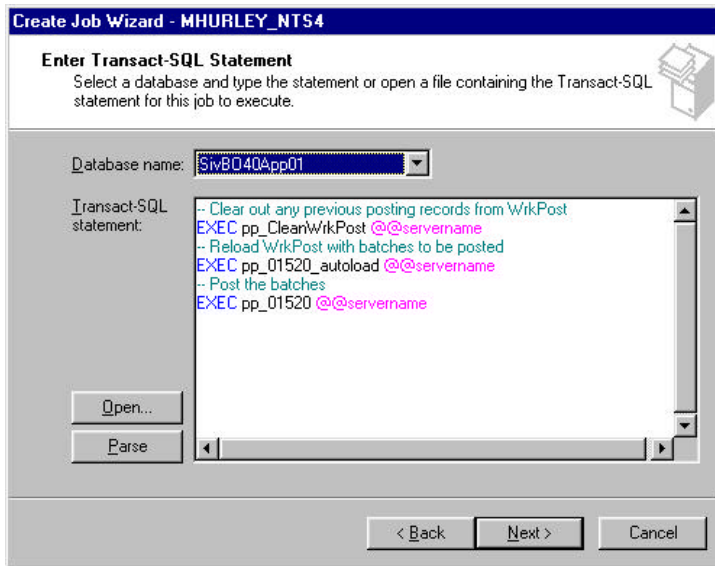


Figure 3 Create Job Wizard - Enter Transact-SQL Statement

The dialog in Figure 4 illustrates scheduling the job to run daily at 1:00am.

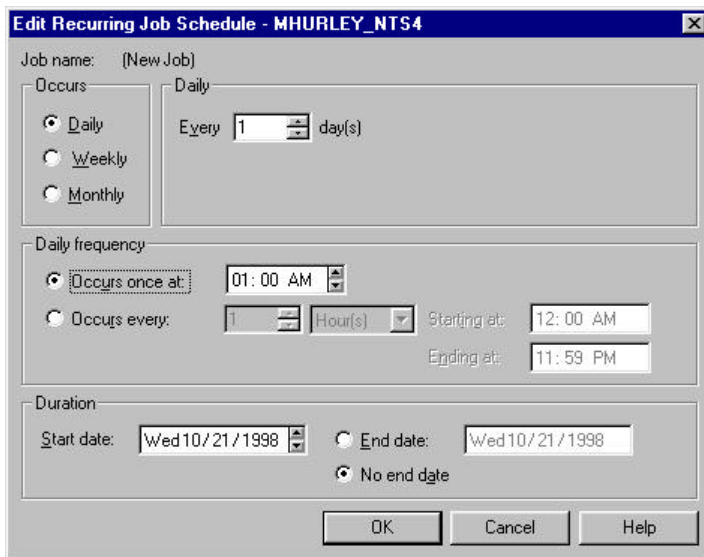


Figure 4 Create Job Wizard – Edit Recurring Job Schedule

Upon completion, notifications containing job status can be sent to specific operators.

Figure 5 shows the next step in the wizard, which specifies that an email be sent to the operator called “SYSADMIN”.

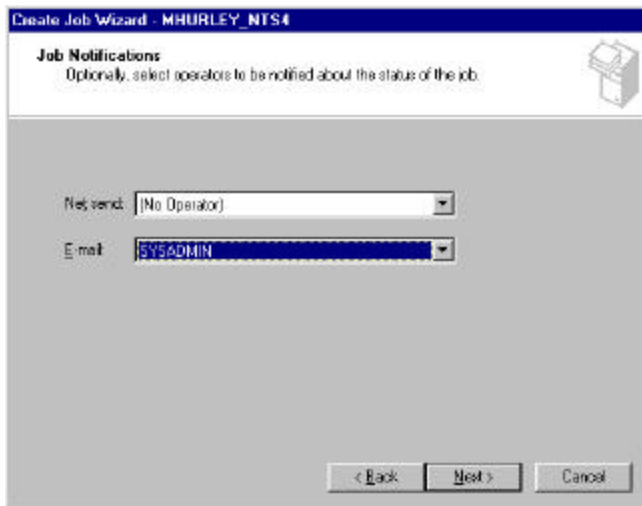


Figure 5 Create Job Wizard - Job Notifications

Figure 6 shows the final step in this wizard, which allows the user to name the job and displays a description of the job that was created by the wizard.

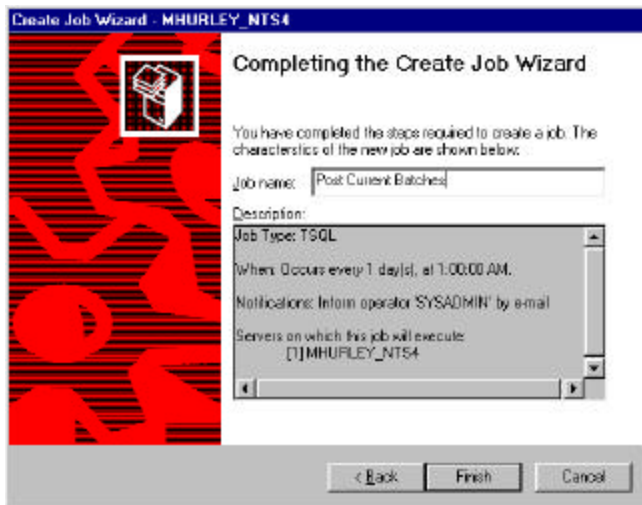


Figure 6 Completing the Create Job Wizard

Upon clicking the “Finish” button in this example, a job called “Post Current Batches” is created in the “SivBO40App01” database and scheduled to run nightly at 1:00am

Backup and Restore

SQL Server 7.0 offers faster backup and recovery performance as well as a new incremental backup / restore feature. A feature called incremental backups save only those data pages that have changed since the last database backup (as opposed to the entire database or just the transaction log). Restoring an incremental backup often saves time over transactional backups because the server does not have to roll all the transactions forward. It simply restores the changed pages to the state they were in at the point of the backup.

These backup features offer Solomon IV users an efficient, simple method of backing up their data while the databases are still online. Additionally, the backup process can be completely automated through the backup wizard or the database maintenance plan wizard, ensuring a valid backup as frequently as the site requires.

In order to keep all company data synchronized between system and application databases all Solomon databases should be backed up at the same time. Figure 7 shows an example how easy it is to create a Database Maintenance Plan to back up all Solomon databases at the same time. In this dialog, the database maintenance plan is created for the system and two application databases.

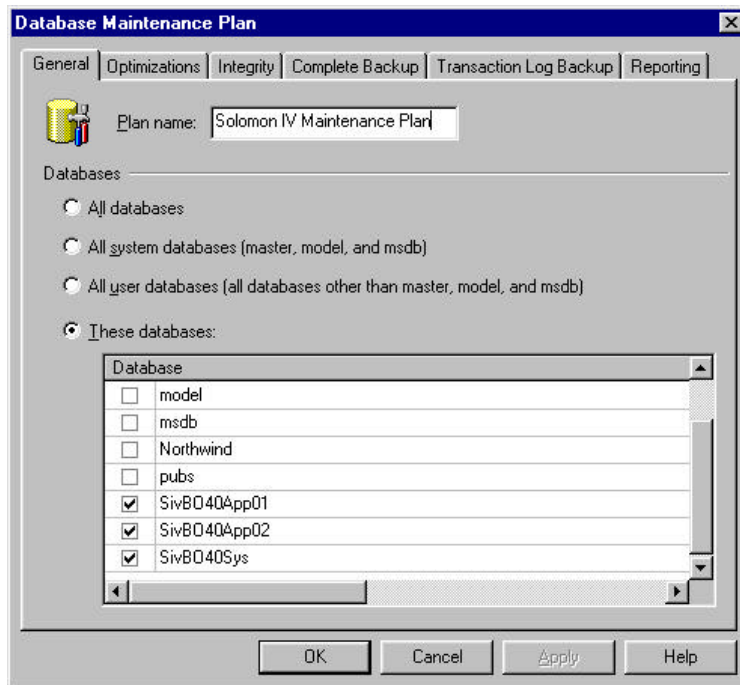


Figure 7 Database Maintenance Plan – Database Selection

The Database Maintenance Plan can schedule complete and transaction log backups to run automatically. Figure 8 shows an example of a backup plan for all selected databases. In this case, a complete backup is performed every Sunday at 2:00am. Each database is backed up to a separate sub-directory and backups that are more than 4 weeks old are automatically removed.

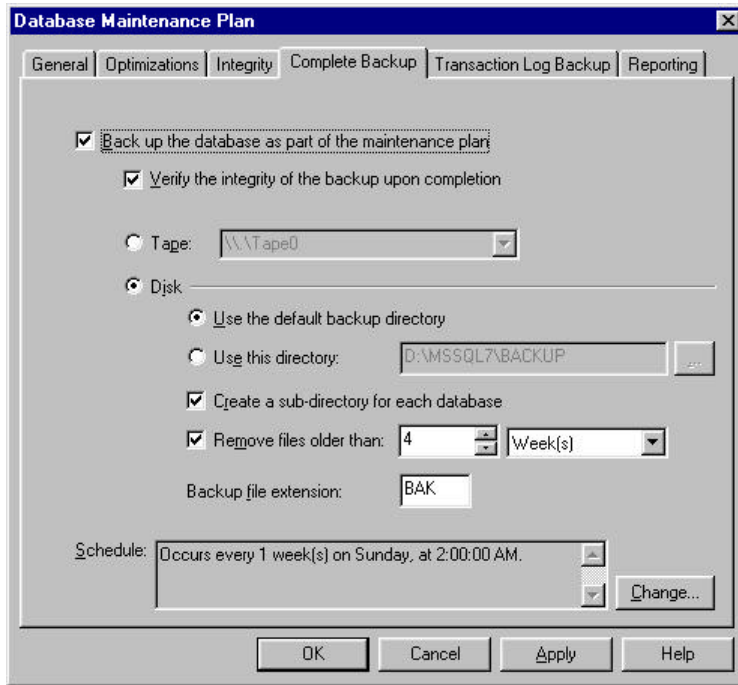


Figure 8 Database Maintenance Plan - Complete Backup

Data Management Objects

SQL Server 7.0 offers Data Management Objects (DMO), a set of COM objects that can be used to programmatically control SQL Server and the database schema. Solomon IV's database installation utility relies on DMO to create database schema, populate tables with initial values, and perform updates to the schema as new applications are added to the system.

Create Database (98.290.00)

Database Definition

Database Type: Application

Destination SQL Server Name: MHURLEY_NTS4

Database Name: AppDb

DB Size(MB): 800

Create Database Transaction Log?

Transaction Log Size(MB): 200

File Locations

Path of Configuration File: D:\Program Files\Solomon IV\BackOffice\DB\DBBuild.ini

Server Location of Database Device Files: E:\MSSQL7\Data

Build Database Update Database Create Backup Close

Connected to MHURLEY_NTS4 (7.0) as sa

Figure 9 Create Database (98.290.00)

Online Analytical Processing Services

Besides simply storing operational data for later processing, retrieval, or archival, businesses today are attempting to gain advantages by using this data for improved strategic decision making and analysis. Data warehousing involves consolidating and summarizing an organization's entire set of heterogeneous data sources into a central location, or data mart, for later analysis. SQL Server 7.0 includes a number of features that create an excellent solution for an organization's data mart.

Because Solomon IV for BackOffice is built on SQL Server using standard features, such as primary keys in all tables and common views of company data, the Solomon IV operational databases make a great source of data for a company's data mart. Using Solomon IV and other corporate data sources to populate a data mart offers tremendous value by allowing sophisticated analysis of all of an organization's data.

Data Transformation Services

Data Transformation Services (DTS) is a utility that allows import, export, and transformation of any heterogeneous data source supported by ODBC or OLE DB. Since Solomon's data resides in SQL Server databases, which are supported by OLE DB, DTS can be used to transform and consolidate Solomon's operational data into a separate data mart.

Online Analytical Processing

SQL Server 7.0 provides integrated Online Analytical Processing (OLAP) Services that support sophisticated multi-dimensional queries against relational data. The primary OLAP object is called a cube, which is a multidimensional representation of detail and summary data. OLAP Services provides a Cube Wizard and Cube Editor to easily create the dimensions and details of a cube.

Replication

Enhanced replication features included with SQL Server 7.0 make it easy to replicate copies of data to the data mart. As with DTS, since Solomon IV data resides on SQL Server, it is very easy to define replication jobs that copy Solomon operational data to data mart databases for consolidation and analysis.

Conclusion

The Solomon IV for BackOffice suite of applications offers an excellent business solution optimized for Microsoft SQL Server. SQL Server 7.0 offers Solomon IV many benefits including improved performance, lower administrative overhead, and greater scalability. SQL Server's rich feature set offers each site the ability to vastly extend the functionality of Solomon IV by creating customized data marts using Solomon's data. The combination of Solomon IV for BackOffice and SQL Server 7.0 is truly the foundation on which to build your business.



Corporate & International Headquarters

P.O. Box 414

Findlay, OH 45840-0414 USA

419-424-0422 FAX 419-424-3400

International Fax 419-422-2044

Email: sales@solomon.com

www.solomon.com

USA & Canada 800-4SOLOMON

(800-476-5666)

International +1 419-424-1422

Solomon Software, the Solomon logo, Solomon III, Solomon IV and Accounting For The Changing Needs Of Business are trademarks of Solomon Software, Inc. Window, Windows for Workgroups, Windows 95, Windows NT Workstation, Windows NT Server and Microsoft SQL Server are products of Microsoft Corporation. Other products mentioned within are trademarks of their respective companies. ©1998 Solomon Software, Inc. All Rights Reserved. Printed in the USA.