

## CHAPTER 2

# SQL Remote Replication Concepts

About this chapter      This chapter introduces the concepts, design goals, and features of SQL Remote.

### Contents

<b>Topic</b>	<b>Page</b>
Introduction to data replication	10
SQL Remote concepts	13
SQL Remote components	17
Publications and subscriptions	20
SQL Remote features	22
Some sample installations	24

## Introduction to data replication

Data replication is the sharing of data among physically distinct databases. Changes made to shared data at any one database are replicated to the other databases in the replication setup. The SQL Remote data replication system enables replication of data among Adaptive Server Anywhere or Adaptive Server Enterprise databases.

### Benefits of data replication

- Data availability** One of the key benefits of a data replication system is that data is made available locally, rather than through potentially expensive, less reliable, and slow connections to a single central database. Data is accessible locally even in the absence of any connection to a central server, so that you are not cut off from data in the event of a failure of a long-distance network connection.
- Response time** Replication improves response times for data requests for two reasons. Requests are processed on a local server without accessing some wide area network, so that retrieval rates are faster. Also, local processing offloads work from a central database server so that competition for processor time is decreased.

### Challenges for replication technologies

- Any replication technology must address several challenges that arise as a result of the increased flexibility permitted by replication.
- Transactional integrity** One of the challenges of any replication system is to ensure that each database retains transactional integrity at all times. Today's replication systems, such as Sybase Replication Server and SQL Remote, replicate portions of the transaction log in such a way that transactions are replicated atomically: either a whole transaction is replicated, or none of it is replicated. This ensures transactional integrity at each database in the setup.
- Data consistency** Another challenge to replication systems is to maintain data consistency throughout the setup. SQL Remote maintains a **loose consistency** in the setup as a whole: that is, all changes are replicated to each site over time in a consistent manner, but because of the time lag different sites may have different copies of data at any instant.

## Sybase replication technologies

Sybase provides two replication technologies; SQL Remote and Replication Server.

- ◆ **SQL Remote** is designed for two-way replication involving a consolidated data server and large numbers of remote databases, typically including many mobile databases. Administration and resource requirements at the remote sites are minimal, and a typical time lag between the consolidated and remote databases is on the order of hours.
- ◆ **Replication Server** is designed for replication among relatively small numbers of data servers, with a typical time lag between primary data and replicate data of a few seconds, and generally with an administrator at each site.

### Choosing a replication technology

SQL Remote is designed for replication installations with the following requirements:

- ◆ **Large numbers of databases** SQL Remote is designed to support potentially large number of remote databases. It can support thousands of remote databases in a single installation.
- ◆ **Occasionally connected** Remote databases that are occasionally connected or indirectly connected to the network on which the server is running.
- ◆ **High latency** High latency means a long lag time between data being entered at one database and being replicated to each database in the installation. With SQL Remote, replication messages are sent typically at periods of hours or days.
- ◆ **Low volume** As replication messages are delivered occasionally, high volume can lead to very large numbers of messages being sent. SQL Remote is best suited to systems with a relatively low volume of replicated data per database.
- ◆ **Homogeneous databases** SQL Remote supports Adaptive Server Enterprise and Adaptive Server Anywhere databases. Each database in the system must have a very similar schema.

Replication Server is designed for replication installations with the following requirements:

- ◆ **Small numbers of databases** Replication Server is designed to support replication among servers, with installations typically involving under one hundred servers.

- ◆ **Continuously connected** Connections between primary sites and replicate sites may be over a wide area network, but Replication Server is designed for situations where there is a near-continuous connection path for data exchange among the servers in the installation.
- ◆ **Low latency** Low latency means a short lag time between data being entered at one database and being replicated to each database in the installation. With Replication Server, replication messages are sent typically within seconds of being entered at a primary site.
- ◆ **High volume** With near-continuous connections and high performance, Replication Server is designed for a high volume of replication messages.
- ◆ **Heterogeneous databases** Replication Server supports several leading DBMS's, and allows mapping of object names during replication, so that support for heterogeneous databases is provided.

## Sybase SQL Remote

Sybase SQL Remote technology is provided in two forms:

- ◆ **SQL Remote for Adaptive Server Anywhere** Enables replication between a consolidated Adaptive Server Anywhere database and a large number of remote databases.
- ◆ **SQL Remote for Adaptive Server Enterprise** Enables replication between a consolidated Adaptive Server Enterprise database and a large number of remote databases.

This book describes both of these technologies.

In a SQL Remote installation, you must have properly licensed SQL Remote software at each participating database.

## SQL Remote concepts

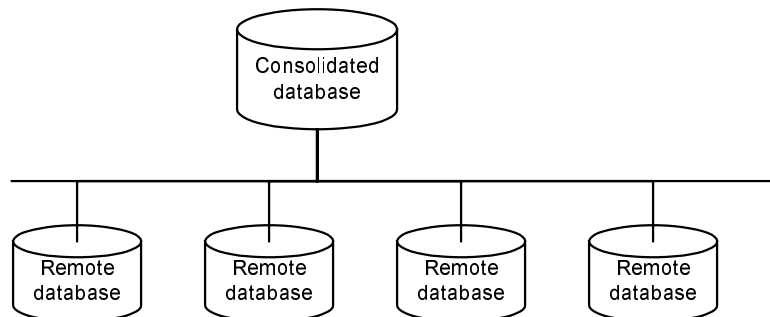
This section introduces concepts and terms used throughout the book.

### Consolidated and remote databases

SQL Remote provides data replication between a **consolidated database** and a set of **remote databases**.

A **consolidated database** is a database that contains all the data to be replicated. A **remote database** is a database that may be running at the same site as the consolidated database or at a physically distant site.

The figure shows a schematic illustration of a small SQL Remote installation.



#### Remote users

A replication installation includes many copies of the information in a database. Each copy is a physically separate database on a separate computer. All remote copies must stay consistent with the consolidated database.

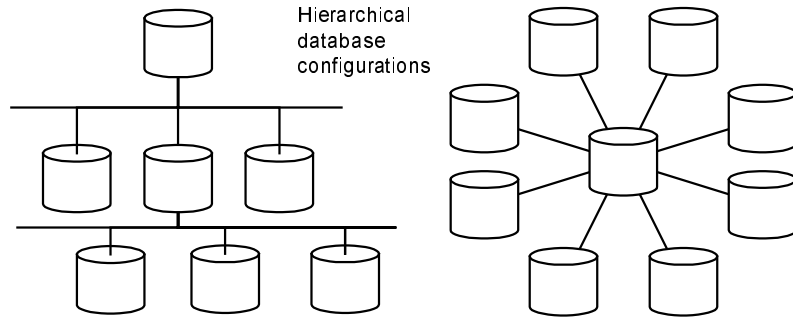
The entire replication setup may be considered a single **dispersed database**, with the master copy of all data being kept at the consolidated database.

Each remote site that submits replications to the consolidated database is considered to be a **remote user** of the consolidated database. In the case that a remote site is a multi-user server, the entire site is considered to be a single remote user of the consolidated database.

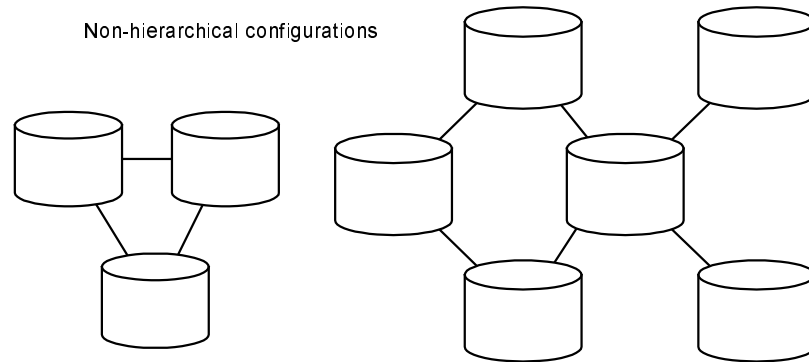
### Hierarchical database configurations

SQL Remote supports **hierarchical configurations** of databases; it does not support peer-to-peer replication or other non-hierarchical configurations.

For any two databases in a hierarchical configuration, one is always above or below the other in the hierarchy.



For databases in a non-hierarchical configuration, there is not any well-defined notion of above or below.



In a SQL Remote installation, each database contains all or a subset of the data replicated by the database above it in the hierarchy.

Remote databases can contain tables that are not present at the consolidated database, as long as they are not involved in replication.

## Two-way replication

SQL Remote replication is two-way: changes made at the consolidated database are replicated to remote databases, while changes made at remote databases are replicated to the consolidated database, and thence to other remote databases.

## Message-based replication

SQL Remote exchanges data between databases using **messages**. This allows replication between databases that have no direct connection: an occasional message-based connection such as e-mail or a periodic dial-up link is sufficient.

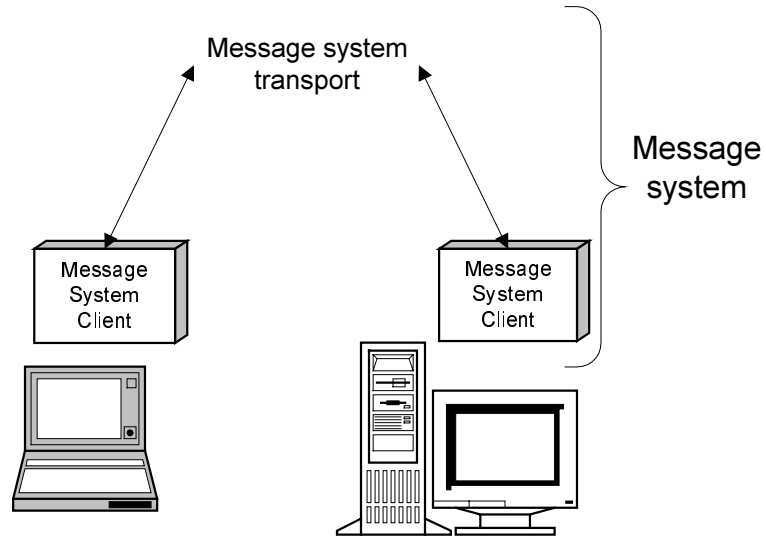
In message-based communications, each message carries its destination address and other control information, so that no direct connection is needed between applications exchanging information. For example, an e-mail message contains the destination address; there is no direct connection between the sending server and the recipient.

### Message services use store and forward methods

Just as session-based client/server applications rely on network communication protocol stacks, such as TCP/IP or Novell NetWare's IPX, so message-based applications rely on message services such as Internet Simple Mail Transfer Protocol (SMTP), Microsoft's Messaging API (MAPI), Lotus' Vendor Independent Messaging (VIM), or a simple shared file link.

Message services use **store-and-forward** methods to get each message to its destination: for example, e-mail systems store messages until the recipient opens their mail folder to read their mail, at which time the e-mail system forwards the message.

Building a replication system on top of a message system means that SQL Remote does not need to implement a store-and-forward system to get messages to their destination. Just as session-based client/server applications do not implement their own protocol stacks to pass information between client and server, so SQL Remote uses existing message systems to pass the messages.



**Guaranteed delivery**

Unlike some session-based communications, many message-based systems do not guarantee that messages reach their destination, or that messages are received in the same order they were sent.

SQL Remote incorporates a protocol to guarantee reception of replication updates in the correct order.



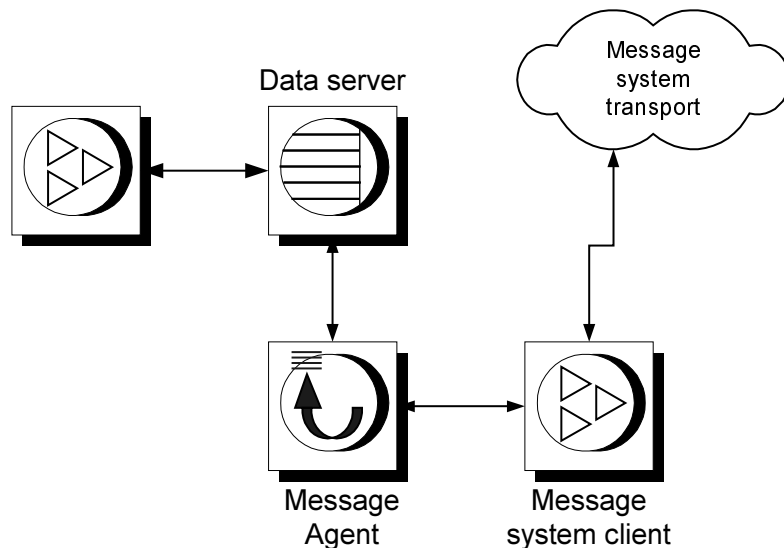
## SQL Remote components

The following components are required for SQL Remote:

- ◆ **Data server** An Adaptive Server Anywhere or Adaptive Server Enterprise database management system is required at each site to maintain the data.
- ◆ **Message Agent** A SQL Remote Message Agent is required at the consolidated site and at each remote site to send and receive SQL Remote messages.

The Message Agent connects to the data server by a client/server connection. It may run on the same machine as the data server or on a different machine.

- ◆ **Database extraction utility** The extraction utility is used to prepare remote databases from a consolidated database, during development and testing, and also at deployment time.
- ◆ **Message system client software** Each computer involved in a setup must have a message system client installed. SQL Remote does support a file-sharing "message system", which does not require client software in addition to the SMTP/POP, MAPI, and VIM message systems under some operating systems.
- ◆ **Client applications** The applications that work with SQL Remote databases are standard client/server database applications.



## The data server

The data server may be an Adaptive Server Enterprise or an Adaptive Server Anywhere server. At the remote site the data server is commonly an Adaptive Server Anywhere personal server, but can also be an Adaptive Server Enterprise or Adaptive Server Anywhere server.

## Client applications

Client applications work with the data in the database. Client applications use one of the client/server interfaces supported by the data server:

- ◆ For Adaptive Server Anywhere, the client application may use ODBC, Embedded SQL, or Sybase Open Client to work with Adaptive Server Anywhere.
- ◆ For Adaptive Server Enterprise, the client application may use one of the Sybase Client Server interfaces, ODBC, or Embedded SQL.

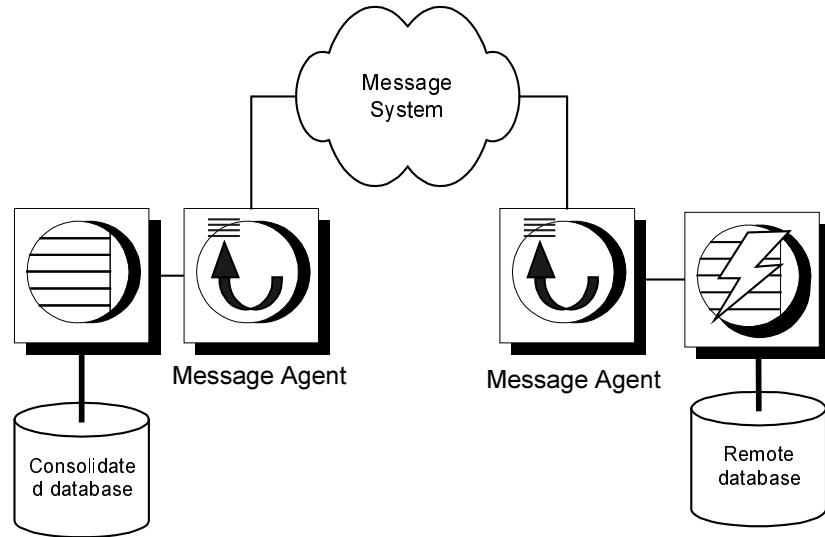
Client applications do not have to know if they are using a consolidated or remote database. From the client application perspective, there is no difference.

## The Message Agent

The SQL Remote **Message Agent** sends and receives replication messages. It is a client application that sends and receives messages from database to database. The Message Agent must be installed at both the consolidated and at the remote sites.

For Adaptive Server Anywhere, the Message Agent is a program called *dbremote.exe* on PC operating systems, and *dbremote* on UNIX.

For Adaptive Server Enterprise, the Message Agent is a program called *ssremote.exe* on PC operating systems, and *ssremote* on UNIX.



### Message system client

If you are using a shared file message system, no message system client is needed.

If you are using an e-mail or other message system, you must have a message system for that client in order to send and receive messages.

# Publications and subscriptions

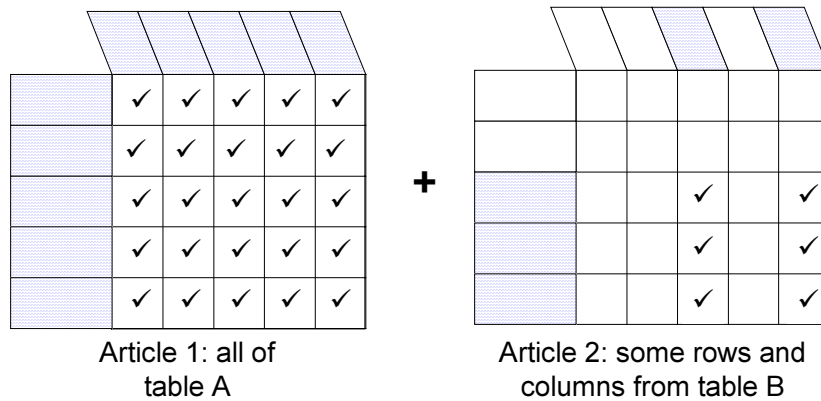
Data is organized into publications

The data that is replicated by SQL Remote is arranged in **publications**. Each database that shares information in a publication must have a **subscription** to the publication.

The **publication** is a database object describing data to be replicated. Remote users of the database who wish to receive a publication do so by **subscribing** to a publication.

A publication may include data from several database tables. Each table's contribution to a publication is called an **article**. Each article may consist of a whole table, or a subset of the rows and columns in a table.

## A two-table publication



Periodically, the changes made to each publication in a database are replicated to all subscribers to that publication. These replications are called publication **updates**.

Messages are always sent both ways

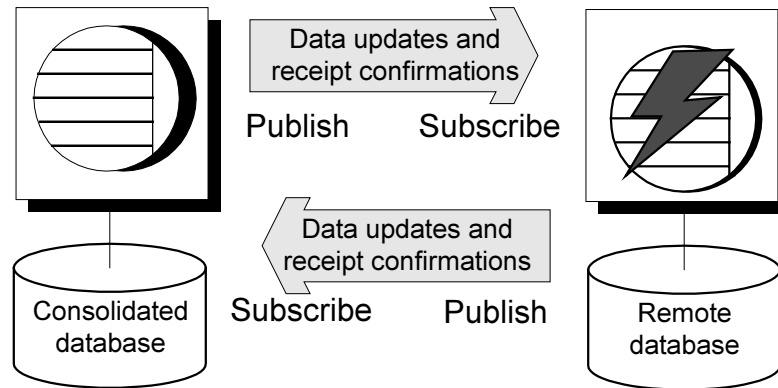
Remote databases subscribe to publications on the consolidated database so that they can receive data from the consolidated database. To do this, a **subscription** is created at the consolidated database, identifying the subscriber by name and by the publication they are to receive.

SQL Remote always involves messages being sent two ways. The consolidated database sends messages containing publication updates to remote databases, and remote databases also send messages to the consolidated database.

For example, if data in a publication at a consolidated database is updated, those updates are sent to the remote databases. And even if the data is never updated at the remote database, **confirmation messages** must still be sent back to the consolidated database, to keep track of the status of the replication.

Both databases subscribe

Messages must be sent both ways, so not only does a remote database subscribe to a publication created at the consolidated database, but the consolidated database must subscribe to a corresponding publication created at the remote database.



When remote database users modify their own copies of the data, their changes are replicated to the consolidated database. When the messages containing the changes are applied at the consolidated database the changes become part of the consolidated database's publication, and are included in the next round of updates to all remote sites (except the one it came from). In this way, replication from remote site to remote site takes place via the consolidated database.

Synchronizing a remote database

When a subscription is initially set up, the two databases must be brought to a state where they both have the same set of information, ready to start replication. This process of setting up a remote database to be consistent with the consolidated database is called **synchronization**. Synchronization can be carried out manually, but the database extraction utility automates the process. You can run the extraction utility from Sybase Central or as a command-line utility.

The appropriate publication and subscription are created automatically at remote databases when you use the SQL Remote database extraction utility to create a remote database.

## SQL Remote features

The following features are key to SQL Remote's design.

**Support for many subscribers** SQL Remote is designed to support replication with many subscribers to a publication.

This feature is of particular importance for mobile workforce applications, which may require replication to the laptop computers of hundreds or thousands of sales representatives from a single office database.

**Transaction log-based replication** SQL Remote replication is based on the transaction log. This enables it to replicate only changes to data, rather than all data, in each update. Also, log-base replication has performance advantages over other replication systems.

The transaction log is the repository of all changes made to a database. SQL Remote replicates changes made to databases as recorded in the transaction log. Periodically, all committed transactions in the consolidated database transaction log belonging to any publication are sent to remote databases. At remote sites, all committed transactions in the transaction log are periodically submitted to the consolidated database.

By replicating only committed transactions, SQL Remote ensures proper transaction atomicity throughout the replication setup and maintains a consistency among the databases involved in the replication, albeit with some time lag while the data is replicated.

**Central administration** SQL Remote is designed to be centrally administered, at the consolidated database. This is particularly important for mobile workforce applications, where laptop users should not have to carry out database administration tasks. It is also important in replication involving small offices that have servers but little in the way of administration resources.

Administration tasks include setting up and maintaining publications, remote users, and subscriptions, as well as correcting errors and conflicts if they occur.

**Economical resource requirements** The only software required to run SQL Remote in addition to your Adaptive Server Anywhere or Adaptive Server Enterprise DBMS is the Message Agent, and a message system. If you use the shared file link, no message system software is required as long as each remote user ID has access to the directory where the message files are stored.

Memory and disk space requirements have been kept moderate for all components of the replication system, so that you do not have to invest in extra hardware to run SQL Remote.

**Multi-platform support** The Message Agent for Adaptive Server Anywhere runs on Windows 95, Windows NT, Windows 3.x, NetWare, UNIX, and OS/2 platforms.

The Message Agent for Adaptive Server Enterprise runs on Windows NT, and UNIX.

The UNIX Message Agent supports only the shared file link.

## **Some sample installations**

While SQL Remote can provide replication services in many different environments, its features are designed with the following characteristics in mind:

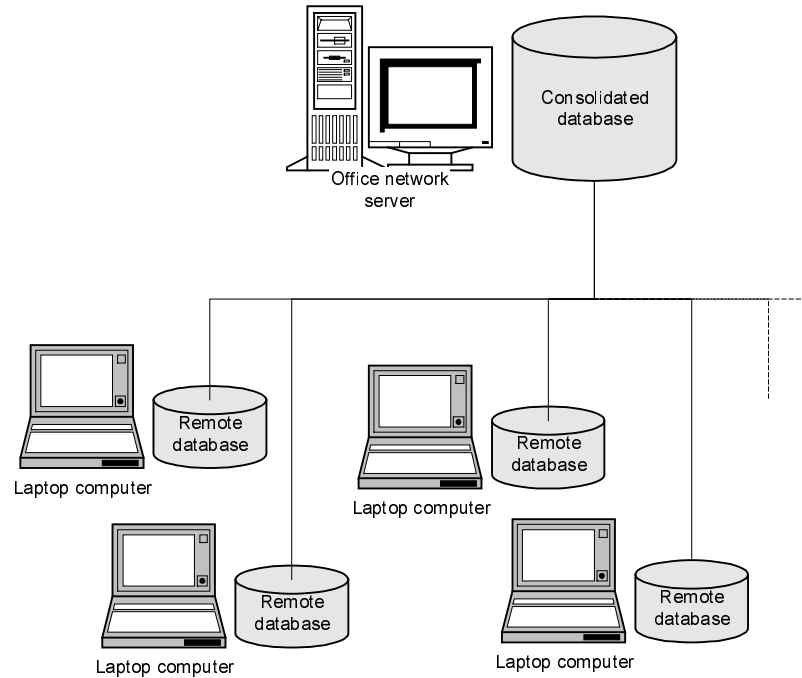
- ◆ SQL Remote should be a solution even when no administration load can be assigned to the remote databases, as in mobile workforce applications.
- ◆ Data communication among the sites may be occasional and indirect: it need not be permanent and direct.
- ◆ Memory and resource requirements at remote sites are assumed to be at a premium.

The following examples show some typical SQL Remote setups.

### **Server-to-laptop replication for mobile workforces**

SQL Remote provides two-way replication between a database on an office network and personal databases on the laptop computers of sales representatives. Such a setup may use an e-mail system as a message transport.





The office server may be running a server to manage the company database. The Message Agent at the company database runs as a client application for that server.

The laptop computers may be running Windows 95 or Windows NT, and each sales representative has an Adaptive Server Anywhere personal server to manage their own data.

While away from the office, a sales representative can make a single phone call from their laptop to carry out the following functions:

- ◆ Collect new e-mail.
- ◆ Send any e-mail messages they have written.
- ◆ Collect publication updates from the office server.
- ◆ Submit any local updates, such as new orders, to the office server.

The updates may include, for example, new specials on the products the sales representative handles, or new pricing and inventory information. These are read by the Message Agent on the laptop and applied to the sales rep's database automatically, without requiring any additional action on the sales representative's part.

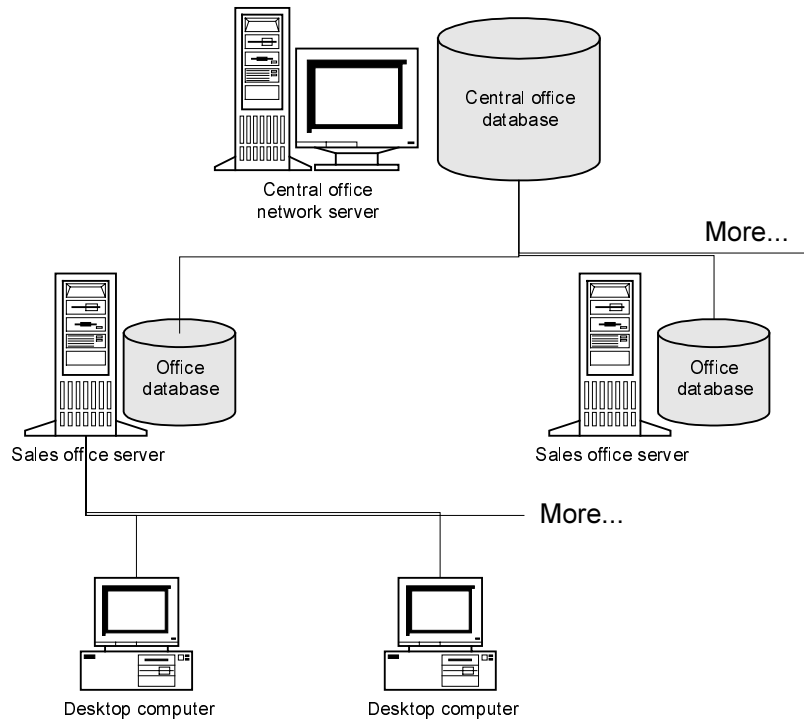
The new orders recorded by the sales representative are also automatically submitted to the office without any extra action on the part of the sales representative.

## Server-to-server replication among offices

SQL Remote provides two-way replication between database servers at sales offices or outlets and a central company office, without requiring database administration experience at each sales office beyond the initial setup and that required to maintain the server.

SQL Remote is not designed for up-to-the-minute data availability at each site. Instead, it is appropriate where data can be replicated at periods of an hour or so.

Such a setup may use an e-mail system to carry the replication, if there is already a company-wide e-mail system. Alternatively, an occasional dial-up system and file transfer software can be used to implement a FILE message system.



SQL Remote is easy to configure to allow each office to receive their own set of data. Tables that are of office interest only (staff records, perhaps, if the office is a franchise) may be kept private in the same database as the replicated data.

Layers can be added to SQL Remote hierarchies: for example, each sales office server could act as a consolidated database, supporting remote subscribers who work from that office.

