

谈谈VFP和SQLSERVER搭配做C_S系统 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/251/2021_2022__E8_B0_88_E8_B0_88VFP_E5_c97_251343.htm

1.CLIENT/SERVER 到底是什么? C/S属于2-TIER系统，适合于中小型应用系统。大系统一般都用3-TIER了。打个比方单机数据库系统(比如VFP), 相当于前店后库.店里需要什么东西,得自己去库房找,库房管理也是有你自己进行. 而C/S系统下,店和库是相对独立的,有一个专门的库房管理(数据SERVER),店里需要什么,按照手续把单子给 库房管理人员,由他们去操作. 因此可以看出C/S的优点: 支持多用户. 更有效的数据管理,数据安全和可靠得多.远程使用数据. 如果你是单机使用当然没有太大必要使用C/S了. 如果用VFP本身做C/S系统, 效果不是很明显,因为VFP的数据管理功能不是很强,比如加锁解锁都需要程序来操作. 我主要说的是VFP/SQL SERVER系统.

2. SQL SERVER SQL SERVER是微软发布的RDBMS(关系数据库管理系统), ORACLE, INFOMIX,POWERBUILD,也都是类似的系统. 他们就相当于仓库的管理系统,但功能不仅仅是数据管理. 微软的数据库产品从功能和规模由小到大排列依次是: ACCESS, VFP, SQL SERVER. 稍微大一点的系统,SQL SERVER是需要专人管理的,这就是DBA (DATABASE ADMINISTRATOR)的位置.现在北美人才市场上, DBA的工作比程序员的工作好找些. 工资比普通程序员略高, 工作稳定性也强一些. SQL SERVER是一套大的软件系统,可以安装在专门的NT数据SERVER上,也有个人版可以安装在WIN95/98上, 主要是为了咱们这些程序员方便测试. 它的功能主要有这么几块: 数据库的管理和维护,用户/安全管理,

数据的发布/转换. 3. 用VFP/SQL SERVER做C/S系统 VFP和SQL SERVER的搭配应该还是比较完美的, 但因为VB才是微软的主流产品(VB一直是微软的,FOXPRO是后来才买的,后娘生的), 所以微软从来没有大力推荐或者宣传过VFP, 而是把VFP独有的数据库技术融合到自己的其它产品里了: SQL SERVER, ODBC, OLE DB/ADO. 最近甚至把VFP从VISUAL STUDIO里独立出来了. VFP和SQL SERVER的交流,可以通过3种方式进行: A. 远程视图 B. ADO控件 C. SPT (SQL PASS THROUGH) 3种方式各有优缺点, 而我最偏好第3种. 远程视图和ADO都在VFP和SQL SERVER中间加了一个层次, 而这层次就相当于一个黑箱,你不清楚它们到底怎么操作数据的,只需按照它们的规则进行设置,使用相应的命令就可以. 反正我作为程序员,喜欢清楚地知道我的程序每一步都在干什么,所以我喜欢用SPT,也从不用向导来建立表单报表一类的. SPT技术是通过VFP的函数SQLCONNECT()来和SQL SERVER建立连接, 然后用SQLEXP()函数把要执行的SQL命令送到SQL SERVER上去执行, 所有对数据的操作都是通过这些SQL命令来进行的. 每一个细节都由自己的代码来控制. 4. VFP数据的升迁 VFP的数据库,可以直接用VFP自带的升迁向导转到SQL SERVER上,但必须把所有的表都放到数据库里,自由表不能直接升迁, 而实际上升迁上去的又只是表,数据库本身不能转到SQL SERVER里. 感觉怪怪的. 也许是为了升迁索引或者关系吧. 要把VFP数据升迁到SQL SERVER上,必须先先在SQL SERVER里建立相应的数据库. SQL SERVER的管理是通过ENTERPRISE MANAGER (EM)来进行的. 通过它建立数据库, 每个数据库里可以包含数据表,用户/用户群,视图,存储过程等等. 至于改用SQL SERVER

后,原来的VFP程序是肯定需要修改的,而且是比较大的改动.

***** 谈谈VFP和SQL SERVER搭配做C/S系统 (二) --
动态设置ODBC连接 1. 设置ODBC数据源 VFP和SQL
SERVER的连接是通过ODBC或者OLE DB(ADO)来进行的。可以打开控制面板的ODBC数据源进行设置。但很多情况下，我们希望能在程序里动态设置数据源，一来可以不用去为每个用户的机器手动设置，二来为了数据的安全性，使用完后，希望把ODBC数据源删掉。下面的例子是用API函数来设置和删除ODBC数据源。函数名字是 SQLConfigDataSource，其中第二项参数是数字(1-增加 2-修改 3-删除) DECLARE
INTEGER SQLConfigDataSource IN odbccp32 INTEGER,
INTEGER, STRING, STRING InWindowHandle=0
lcODBCDriver= ' SQL Server ' amp.DRIVER类型
lcODBCName= ' SharedData ' amp.数据源名字 lcODBCDesc=
' Shared Data Source ' amp.数据源描述 lcODBCServer=
' DEVSQL ' amp.SQL SERVER名字 lcODBCDatabase=
' Shared ' amp.要连接的数据库名字 **先试图修改已有的ODBC，如果不存在，返回0。
lreturn=SQLConfigDataSource(InWindowHandle, 2,
lcODBCDriver, . ' DSN= ' lcODBCName CHR(0) .
' Description= ' lcODBCDesc CHR(0) . ' Server= '
lcODBCServer CHR(0) . ' Database= ' lcODBCDatabase
CHR(0)) IF lreturn=0 amp.不存在 **添加新的ODBC
lreturn=SQLConfigDataSource(InWindowHandle, 1,
lcODBCDriver, . ' DSN= ' lcODBCName CHR(0) .
' Description= ' lcODBCDesc CHR(0) . ' Server= '

lcoDBCServer CHR(0) . ' Database= ' lcoDBCDatabase
 CHR(0)) IF lreturn=0 amp.失败 MessageBox(' 添加ODBC数据
 源失败 ' ,16, ' BUFFER ') ENDIF ENDIF 2。删除ODBC数据
 源 **用完后，可以在表单DESTROY事件里删除ODBC。如果
 不想重复设置参数，可以把这些参数加到表单作为属性。
 DECLARE INTEGER SQLConfigDataSource IN odbccp32
 INTEGER, INTEGER, STRING, STRING lnWindowHandle=0
 lcoDBCDriver= ' SQL Server ' lcoDBCName= ' SharedData '
 lcoDBCDesc= ' Shared Data Source ' lcoDBCServer=
 ' DEVSQL ' lcoDBCDatabase= ' Shared ' **先修改，或者其
 是否存在 lreturn=SQLConfigDataSource(lnWindowHandle, . 2,
 lcoDBCDriver, . ' DSN= ' lcoDBCName CHR(0) .
 ' Description= ' lcoDBCDesc CHR(0) . ' Server= '
 lcoDBCServer CHR(0) . ' Database= ' lcoDBCDatabase
 CHR(0)) IF lreturn=1 amp.ODBC存在，删除它
 lreturn=SQLConfigDataSource(lnWindowHandle, . 3,
 lcoDBCDriver, . ' DSN= ' lcoDBCName CHR(0) .
 ' Description= ' lcoDBCDesc CHR(0) . ' Server= '
 lcoDBCServer CHR(0) . ' Database= ' lcoDBCDatabase
 CHR(0)) IF lreturn=0 MessageBox(' 删除ODBC源失败,16,
 ' BUFFER ') ENDIF ENDIF 100Test 下载频道开通，各类考试
 题目直接下载。详细请访问 www.100test.com