

如何用VB存取SQLServer中的图像数据 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/646/2021\\_2022\\_\\_E5\\_A6\\_82\\_E4\\_BD\\_95\\_E7\\_94\\_A8V\\_c97\\_646839.htm](https://www.100test.com/kao_ti2020/646/2021_2022__E5_A6_82_E4_BD_95_E7_94_A8V_c97_646839.htm)

MIS SQL Server对图像数据的存储机制和存取方法。针对VB开发工具，介绍了一种通过ADO Field 对象的GetChunk 方法和AppendChunk 方法来存取MIS SQL Server中的图像数据的方法。 ADO Field对象 BLOB 对象 GetChunk和AppendChunk 方法image数据类型医院信息系统 MIS SQL Server 在一个完善的医院信息MIS中，图像数据的存取是必不可少的，比如X光片、CT像片的保存。一方面，这些图像数据在远程诊疗为准确诊断病情提供了重要的依据，另一方面，也为快速查阅病人资料提供了基本条件。图像数据的存取在其它应用系统如GIS中也有广泛的应用。

1、SQL Server中图像数据的存储机制 在MIS SQL Server 中，对于小于 8000 个字节的图像数据可以用二进制型(binary、varbinary)来表示。但通常要保存的一些医学影像图片都会大于 8000个字节。SQL Server提供了一种机制，能存储每行大到 2G的二进制对象（BLOB），这类对象可包括image、text 和ntext三种数据类型。Image数据类型存储的是二进制数据，最大长度是  $2^{31}-1$  (2,147,483,647)个字节[2][3]。BLOB数据在MIS SQL Server系统中的存储方式不同于普通的数据类型，对于普通类型的数据系统直接在用户定义的字段上存储数据值，而对于BLOB类型数据，系统开辟新的存储页面来存放这些数据，表中BLOB类型数据字段存放的仅是一个16个字节的指针，该指针指向存放该条记录的BLOB数据的页面。

2、SQL Server中图像数据的存取 在MIS SQL Server中，当数据小

于 8000 个字节时，可以用普通的SQL操纵语句(SELECT、INSERT、UPDATE、DELETE)来完成对字段的操纵，当数据大于8000个字节时，SQL提供了 WRITETEXT、READTEXT 和UPDATETEXT这三个函数来读取和修改数据。这三个函数的使用方法为：(1) WRITETEXT {table.column text\_ptr}[WITH LOG] table.column为表中的字段，text\_ptr为一个16个字节的指针，data为要写的数据值。可选参数WITH LOG表示是否要写入日志文件中。例: DECLARE @ptrval binary(16) --指针  
SELECT @ptrval = TEXTPTR(img\_ct) FROM zy\_ct WHERE id\_ct = 20010101001  
WRITETEXT zy\_ct.img\_ct @ptrval  
0x024324142342134214213421421454353452341 (2) READTEXT {table.column text\_ptr offset size} [HOLDLOCK] table.column为表中的字段，text\_ptr为一个16个字节的指针，offset 为偏移量，即从第几个字节开始读数据，size为要读的字节数，HOLDLOCK 为在读数据中是否允许其他用户 修改该数据。例：DECLARE @ptrval varbinary(16) SELECT @ptrval = TEXTPTR(img\_ct) FROM zy\_ct WHERE id\_ct = 20010101001  
READTEXT zy\_ct.img\_ct @ptrval 1 25 (3) UPDATETEXT {table\_name.dest\_column\_name dest\_text\_ptr} { NULL | 0delete\_length}[WITH LOG][ inserted\_data | {table\_name.src\_column\_name src\_text\_ptr}]  
table\_name.dest\_column\_name 为要修改的text, ntext, 或 image字段；dest\_text\_ptr为指向其的指针；insert\_offset为偏移量，对于text和image为从第几开始字节开始写，对于ntext为从第几个字符(双字节)开始写；0delete\_length 为从insert\_offset开始删除0delete\_length长度的字节（符），为0时不删除，为NULL

时为删除从insert\_offset开始到结束的所有数据。要插入的数据为inserted\_data为，也可是表table\_name的src\_column\_name字段中指针src\_text\_ptr所指数据。例: DECLARE @ptrval binary(16) SELECT @ptrval = TEXTPTR(img\_ct) FROM zy\_ct WHERE id\_ct = 20010101001 UPDATETEXT zy\_ct.img\_ct @ptrval 16 0x54345 可以看出，这三个函数的使用比较复杂，虽然可以通过生成存贮过程来调用执行，但有一个缺陷是在读取数据时，READTEXT函数读取的数据无法直接传递回前端应用程序。

### 3、VB 6.0中图像数据的存取

VB 6.0的ADO Field对象提供了GetChunk方法和AppendChunk方法来存取BLOB数据[1]，这两个函数实质是通过API调用WRITETEXT、READTEXT和UPDATETEXT这三个函数，简化了调用的方法。

(1) GetChunk和AppendChunk方法介绍

GetChunk方法检索其部分或全部长二进制或字符数据[4]。GetChunk调用返回的数据将赋给“变量”。如果Size大于剩余的数据，则GetChunk仅返回剩余的数据而无需用空白填充“变量”。如果字段为空，则GetChunk方法返回Null。每个后续的GetChunk调用将检索从前一次GetChunk调用停止处开始的数据。但是，如果从一个字段检索数据然后在当前记录中设置或读取另一个字段的值，ADO将认为已从第一个字段中检索出数据。如果在第一个字段上再次调用GetChunk方法，ADO将把调用解释为新的GetChunk操作并从记录的起始处开始读取。Field对象的第一个AppendChunk调用将数据写入字段，覆盖所有现有的数据，随后的AppendChunk调用则添加到现有数据。由于系统资源总是有限的，如果一次读(存)取大量数据，可能会引起服务器、客户机死机或是服

务器的性能大大下降，因此使用这两个函数时，要将图像数据进行分段读写。

(2)程序实现 程序一：写数据函数

```
Public Function AppendBlobFromFile(blobColumn As ADODB.Field, ByVal FileName) As Boolean Dim FileNumber As Integer 文件号 Dim DataLen As Long 文件长度 Dim Chunks As Long 数据块数
```

编辑特别推荐: #0000ff>计算机三级的四个类别看看你适合哪个 #0000ff>全国计算机等考三级数据库模拟试题及参考答案 #0000ff>全国计算机等级考试三级数据库技术训练题 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](http://www.100test.com)